AMSTRAD

ORDINATEUR PERSONNEL PC 1640

GUIDE DE REFERENCE TECHNIQUE

EDITION MICRO APPLICATION

Guide de Référence Technique du PC 1640

(c) Copyright 1988, AMSTRAD Consumer Electronics plc et Locomotive Software Ltd.

Ni l'information, ni le produit décrits dans ce manuel ne peuvent être modifiés ou reproduits totalement ou partiellement, sans l'accord écrit au préalable d'AMSTRAD Consumer Electronics plc ("AMSTRAD").

Les renseignements contenus dans ce manuel sont exacts à la date de parution. Cependant, les produits décrits pouvant être améliorés techniquement après la date de publication, les messages apparaissant à l'écran peuvent différer des messages mentionnés ici.

Traduction : **Jean-Claude BERGER**

Edité par : **MICRO APPLICATION**
13 Rue Sainte Cécile
75009 PARIS

1ère publication 1988 - ISBN : 2-86899-135-1

Code ER : 210

GEM, CPM, DOS PLUS sont des marques déposées de DIGITAL RESEARCH INC.

HERCULES est une marque déposée de HERCULES COMPUTER TECHNOLOGY.

MS-DOS est une marque déposée de MICROSOFT CORP.

AMSTRAD, AMSTRAD PC 1640 et AMSTRAD PC 1512 sont des marques déposées d'AMSTRAD CONSUMER ELECTRONICS PLC.

TABLE DES MATIERES

INTRODUCTION 1

CHAPITRE 1 :

CARACTERISTIQUES TECHNIQUES 3

1.1 Unité centrale de traitement (CPU) 3
1.2 Carte mémoire 4
1.3 Canaux d'E/S de la carte mère 7
1.4 Canaux d'E/S du bus d'extension 8

1.5 Accès direct à la mémoire 9
1.5.1 Registres de page DMA 10
1.5.2 Initialisation DMA 11

1.6 Les Interruptions système 12
1.6.1 Niveaux d'interruptions 13
1.6.2 Initialisation du contrôleur d'interruptions 13
1.6.3 Contrôle de masque NMI 14

1.7 Les timers d'intervalle programmables 14
1.7.1 Configuration du timer 15
1.7.2 Initialisation du compteur 1 15

1.8 Etat et contrôle du système 16
1.8.1 Port B : Contrôle du système 16
1.8.2 Port A : Entrée Status 1 / Codes clavier 17
1.8.3 Port C : Entrée Status 2 19
1.8.4 Ecriture système Status-1 (WSS1) 20
1.8.5 Ecriture système Status-2 (WSS2) 21
1.8.6 Reset système 21

1.9 L'Horloge temps réel (RTC) 22

1.10 Port imprimante parallèle 23
1.10.1 Latch de données de l'imprimante 23
1.10.2 Latch de contrôle de l'imprimante 24
1.10.3 Canal Printer Status 26

1.11 Internal Graphics Adapter (carte IGA intégrée) 27
1.11.1 Affichage couleur en mode texte 29
1.11.2 Affichage texte monochrome 32
1.11.3 Affichage graphique couleur 34
1.11.4 Mode graphique basse résolution (320x200) 34
1.11.5 Mode graphique moyenne résolution (640x200) 36
1.11.6 Mode graphique haute résolution (640x350) 37
1.11.7 Les modes du BIOS IGA 37
1.11.8 Registres de contrôle IGA 39
1.11.9 Initialisation du mode EGA 84
1.11.10 Les registres compatibles CGA 86
1.11.10.1 Le registre CGA Mode Control 87
1.11.10.2 Le registre CGA Color Select 88
1.11.10.3 Le registre CGA Status 89
1.11.10.4 L'émulation du contrôleur d'écran 6845 en mode CGA 90
1.11.10.5 Adressage de l'affichage CRTC 93
1.11.11 Les registres du mode compatible MDA 93
1.11.11.1 Le registre MDA Mode Control 94
1.11.11.2 Le registre MDA Status 94
1.11.11.3 L'émulation du CRTC 6845 95
1.11.12 L'émulation compatible Hercules 96
1.11.12.1 Le registre HMGA Mode Control 96
1.11.12.2 Le registre Status HMGA 99
1.11.12.3 Emulation du CRTC 6845 100

1.12 Le contrôleur de lecteurs de disquettes (FDC) 102
1.12.1 Environnement hardware du FDC 103

1.13 Le port série asynchrone RS232C 104
1.13.1 L'interface série 104
1.13.2 Brochage du connecteur série 105

1.14 L'interface imprimante parallèle 106

1.15 L'interface clavier 107
1.15.1 Les signaux Serial Clock et Serial Data 107
1.15.2 Interface clavier vers unité centrale 108
1.15.3 Unité centrale vers interface clavier 108
1.15.4 Les codes de touches 109
1.15.5 Le Connecteur du clavier 109

1.16 L'interface souris 110
1.16.1 Le connecteur souris 111

1.17 L'interface joystick 113
1.17.1 Le connecteur du joystick 114

1.18 Le connecteur du crayon optique 114
1.19 Les connecteurs d'extension 116
1.20 Le connecteur vidéo 118
1.21 Le connecteur d'alimentation 119
1.22 Configuration des commutateurs d'affichage du PC 1640 121

CHAPITRE 2 :

SYSTEME D'EXPLOITATION 125

2.1 Initialisation et autotest 126

2.2 Autotests de mise en route 132
2.2.1 Procédure de test 133
2.2.2 Méthodes de test 134
2.2.3 Checksum du ROS 135
2.2.4 Test du contrôleur DMA 135
2.2.5 Test du timer d'intervalle programmable 135
2.2.6 Test de l'interface de périphériques programmable 135
2.2.7 Test de l'horloge temps réel 135
2.2.8 Test du circuit de communication série 136
2.2.9 Test du port parallèle 136
2.2.10 Test des registres compteurs X et Y 136
2.2.11 Test de la RAM du système 136

2.2.12 Test du contrôleur d'interruptions programmable 137
2.2.13 Test des lecteurs de disquettes 137
2.2.14 Test de l'interface clavier 137

2.3 Les interruptions du ROS 137
2.3.1 Interruption 2 : erreur de parité (NMI) 138
2.3.2 Interruption 5 : Impression d'écran 138
2.3.3 Interruption 6 : Contrôle des boutons souris 139
2.3.4 Interruption 8 : Interruption horloge système 140
2.3.5 Interruption 9 : Interruption Clavier 141
2.3.5.1 Touches à traitement spécial 144
2.3.6 Interruption 14 : Contrôleur des lecteurs de disquettes 147
2.3.7 Interruption 16 : E/S vidéo compatibles 6845 147
2.3.8 Interruption 16 de l'IGA : E/S vidéo compatibles EGA 161
2.3.9 Interruption 17 : Configuration système 179
2.3.10 Interruption 18 : Taille mémoire 180
2.3.11 Interruption 19 : E/S lecteurs de disquettes 181
2.3.11.1 Paramètres d'appel et registres des fonctions pour disque dur 186
2.3.12 Interruption 20 : E/S série 189
2.3.13 Interruption 21 : Fonctions avancées 194
2.3.14 Interruption 22 : E/S clavier 197
2.3.15 Interruption 23 : E/S imprimante 199
2.3.16 Interruption 24 : Redémarrage système 201
2.3.17 Interruption 25 : Disk Bootstrap 202
2.3.18 Interruption 26 : Horloge système et horloge temps réel 202
2.3.19 Interruption 27 : Break clavier 207
2.3.20 Interruption 28 : Interruption Ticker externe 208
2.3.21 Interruption 29 : Table des paramètres vidéo 208
2.3.22 Interruption 30 : Table des paramètres disques 210
2.3.23 Interruption 31 : Table Matrice vidéo 210

2.4 Variables système en RAM 211
2.5 RAM non volatile 221

2.6 Les messages du ROS 223
2.6.1 Messages d'information ou d'avertissement 223
2.6.2 Messages d'erreurs fatales 225

CHAPITRE 3 :

TABLE DE REFERENCE 227

3.1 Les Liaisons définissant le langage 227
3.2 Utilisation de la mémoire par le processeur 228
3.3 Touches et codes de touches 229
3.4 Registre du processeur de communications asynchrones (8250) 231
3.5 Registre du contrôleur DMA hautes performances (8237A-4) 235
3.6 Mots d'initialisation du contrôleur d'interruptions programmable (8259 A-2) 238
3.7 Les registres du timer programmable (8253) 242

3.8 Registres de l'horloge temps réel HD146818 (RTC) 243
3.8.1 Adresses de l'heure, de la date et de l'alarme 244
3.8.2 Adresses des registres de la RTC 246

3.9 Le contrôleur de lecteurs de disquettes uPD765A 250

LES ANNEXES :

ANNEXE 1 : 271

Interfaces logicielles avec la souris 271

ANNEXE 2 : 287

Configuration du système sous MS-DOS 287

2.1 La commande BREAK 288
2.2 La commande BUFFERS 288
2.3 La commande COUNTRY 289

2.4 La commande DEVICE 289
2.4.1 DRIVER.SYS 290
2.4.2 RAMDRIVE.SYS 291

2.5 La commande DRIVPARM 292
2.6 La commande FCBS 293
2.7 La commande FILES 294
2.8 La commande LASTDRIVE 294
2.9 La commande SHELL 294
2.10 La commande STACKS 295

ANNEXE 3 : 297

Modes d'affichages MS-DOS dépendant du pays 297

ANNEXE 4 : 299

Les connexions RS232C 299

ANNEXE 5 : 309

Câblage Printer Lead (PL-2) 309

ANNEXE 6 : 311

Distribution de l'alimentation du PC 1640 311

ANNEXE 7 : 313

Jeux de caractères en ROM 313

ANNEXE 8 : 315

Codes des touches du clavier 315

ANNEXE 9 : 317

Disposition des touches 317

ANNEXE 10 : **323**

L'éditeur de liens (MS-LINK) 323

10.1 Introduction 323
10.2 Survol de MS-LINK 323
10.3 Définitions de base 326

10.4 Fichiers utilisés par MS-LINK 328
10.4.1 Les extensions des fichiers traités 328
10.4.2 Les extensions des fichiers produits 329
10.4.3 Les fichiers temporaires VM.TMP 329

10.5 Comment lancer MS-LINK 330
10.5.1 Méthode 1 : Invites 331
10.5.2 Méthode 2 : Ligne de commande 332
10.5.3 Méthode 3 : Fichier de réponses 333

10.6 Les caractères de commande 335
10.7 Les invites 336
10.8 Les options MS-LINK 338
10.9 Exemple d'édition de liens 342
10.10 Les messages d'erreur 344

ANNEXE 11 : **349**

L'interpréteur de commandes (COMMAND.COM) 349

ANNEXE 12 : **351**

L'utilitaire de débuggage (DEBUG) 351

12.1 Introduction 351

12.2 Comment lancer DEBUG 351
12.2.1 Méthode 1 : DEBUG 352
12.2.2 Méthode 2 : Ligne de commande 353

12.3 Les commandes DEBUG 353
12.4 Paramètres 355
12.5 Les messages d'erreur 380

ANNEXE 13 : 383

L'utilitaire EXE2BIN 383

ANNEXE 14 : 387

La commande EXIT 387

ANNEXE 15 : 389

L'utilitaire RECOVER 389

ANNEXE 16 : 391

L'utilitaire SHARE 391

ANNEXE 17 : 393

Schémas des ports imprimante et série 393

INDEX 395

PREFACE

Ce manuel de Référence Technique est conçu pour les développeurs de logiciels pour l'Amstrad PC 1640. Utilisé avec le manuel technique Hardware, il sera très précieux aux concepteurs d'extensions matérielles.

Il est tenu pour acquis que le lecteur possède une expérience réelle des systèmes à base de 8086 (ou 8088) et des circuits associés classiquement dans l'architecture d'un ordinateur : contrôleurs DMA, timers programmables, horloges temps réel, etc... Une connaissance des cartes EGA et CGA ainsi que des interfaces pour lecteurs de disquette et ports série et parallèles est aussi indispensable.

Les informations développées ici sont, pour la plupart, exclusives à ce document à l'exception de quelques annexes qui viennent compléter certains sujets du manuel de l'utilisateur du PC 1640 ou du manuel de référence MS-DOS de Microsoft.

Bien que le PC 1640 constitue un sur-ensemble de l'implémentation du standard informatique de fait, ce manuel ne fera pas référence aux différentes améliorations qui y ont été apportées. Les concepteurs devront par conséquent prêter une certaine attention aux problèmes de compatibilité vers le bas qui pourraient se poser en matière de développement pour d'autres machines répondant à ce standard, et devront donc adopter des solutions accessibles au "plus petit dénominateur commun".

INTRODUCTION

Ce manuel constitue une description complète des aspects matériels et logiciels du PC 1640. Les informations d'ordre général à propos de cette machine ainsi que du GEM Desktop et du système d'exploitation qui l'accompagnent figurent dans le manuel de l'utilisateur du PC 1640. Cet ouvrage-ci se propose de satisfaire les besoins des développeurs qui doivent avoir accès à toutes les ressources offertes par le PC 1640.

Notez que toutes les constantes d'adresses de ce document figurent en notation hexadécimale. Les autres valeurs hexadécimales sont repérées par un h minuscule accolé à l'expression. Nous avons choisi cette notation car il est bien évident que l'équivalence binaire de certaines valeurs est plus significative que la quantité proprement dite. Les adresses ne sont pas systématiquement repérées par un h étant donné qu'elles sont implicitement hexadécimales.

CHAPITRE 1

CARACTERISTIQUES TECHNIQUES

1.1 UNITE CENTRALE DE TRAITEMENT

(CPU pour Central Processing Unit)

Le CPU est un microprocesseur 8086-2 présentant une capacité d'adressage de 1 Mo (voir figure 1) cadencé à une fréquence d'horloge de 8 MHz. Le CPU est connecté à un bus de données on-board (intégré à la carte-mère) 16 bits nécessitant quatre cycles de temporisation de 125 nS (T-States) par accès ce qui entraîne un cycle de 500 nS pour la mémoire 16 bits. Le CPU dispose aussi d'un bus intégré 8 bits d'E/S avec une horloge à 4 MHz. Ce dernier est connecté périodiquement à un bus d'extension externe. Les opérations sur le bus 8 bits provoquent automatiquement des temps d'attente de 125 nS comme suit :

Opération	Temps d'attente	Cycle du bus
8 bits (mémoire)	4	1,0 uS
Conversion 16 à 8 bits (mémoire)	12	2,0 uS
8 bits (E/S)	6	1,25 uS
Conversion 16 à 8 bits (E/S)	16	2,5 Us

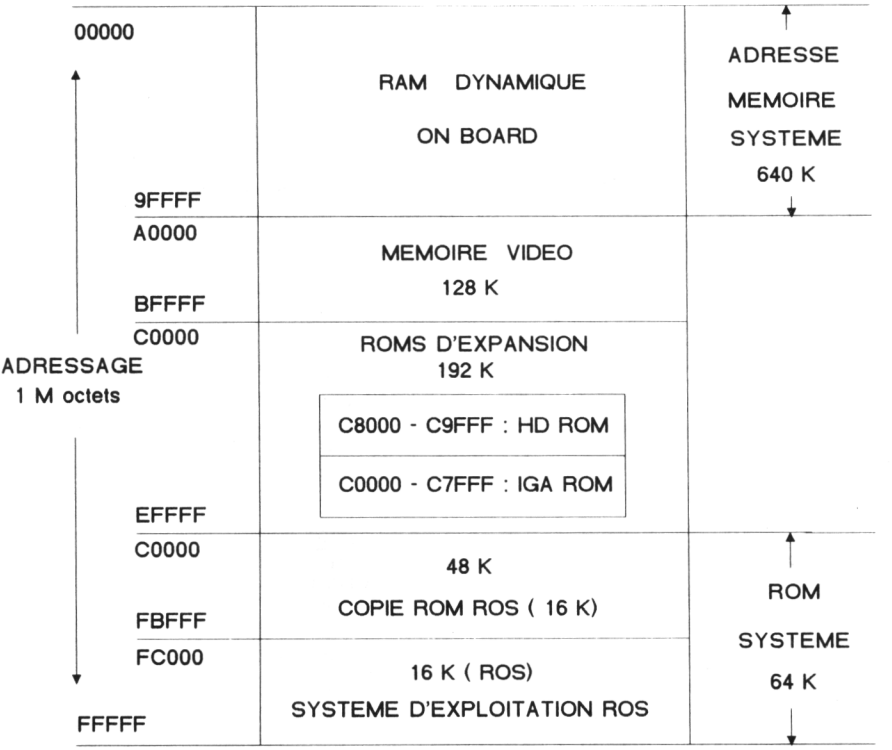
Le CPU est configuré pour fonctionner en mode maximum et son jeu d'instruction peut, optionnellement, être étendu par l'ajout d'un coprocesseur numérique 8087-2. La sortie BUSY du 8087 est directement connectée à l'entrée NOT TEST du 8086.

1.2 CARTE MEMOIRE

L'espace mémoire de la carte mère se compose de 640 K de RAM système avec contrôle de parité et de 16 K de ROM système sans contrôle de parité.

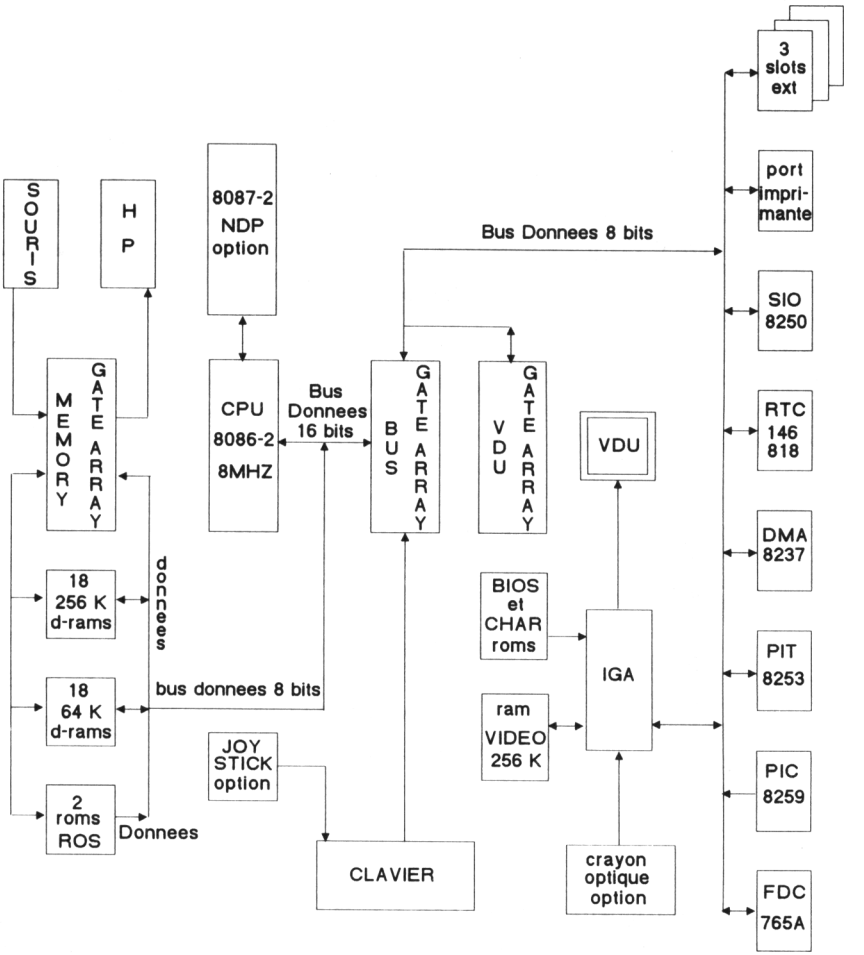
La mémoire utilisateur de 640 K commence à l'adresse CPU 00000 et s'étend jusqu'à 9FFFF.

La zone d'adresses de 128 K entre A0000 et BFFFF est réservée aux tampons de régénération (rafraîchissement) de l'affichage et ne sert pas aux programmes du CPU. La carte vidéo interne IGA (Internal Graphics Adapter) du PC 1640 se sert de l'intégralité des 128 K de RAM vidéo ou de parties (segments) de cette mémoire selon le mode d'affichage sélectionné (voir section 2, Interruption 16 des E/S vidéo). Les cartes vidéo externes peuvent, elles aussi, accéder à cette mémoire et s'en servir comme tampons vidéo.



La zone de 192 K allant de C0000 à EFFFF est réservée à l'adressage des ROM d'extension. Quand l'IGA est actif, il utilise l'espace compris entre C0000 et C1FFF pour son code exécutable (bios) et la zone C2000-C7FFF pour ses fontes (jeux de caractères).

Le contrôleur de disque dur optionnel utilise les adresses C8000 à C9FFF. Les contrôleurs de disques durs additionnels disposent de la zone CA000-CD000. La carte de diagnostic du PC 1640 utilise l'adressage ROM de E0000 à E7FFF.


SOURIS
HP
8087-2
NDP
option
Bus Donnees 8 bits
3
slots
ext
port
impri-
mante
SIO
8250
RTC
146
818
DMA
8237
PIT
8253
PIC
8259
FDC
765A
MEMORY GATE ARRAY
CPU
8086-2
8MHZ
Bus
Donnees
16 bits
BUS GATE ARRAY
VDU GATE ARRAY
VDU
18
256 K
d-rams
donnees
BIOS
et
CHAR
roms
IGA
18
64 K
d-rams
bus donnees 8 bits
ram
VIDEO
256 K
JOY
STICK
option
2
roms
ROS
Donnees
CLAVIER
crayon
optique
option

La ROM système de 16 K est en FC000-FFFFF et contient le logiciel d'exploitation résident (ROS pour Resident Operating System). La zone de 48 K de F0000 à FBFFF est réservée à des extensions de la ROM. Les bits d'adresses de la zone ROS de 16 K sont partiellement décodés de telle sorte que la ROM ROS est répétée 4 fois dans la zone F0000-FFFFF.

1.3 CANAUX D'E/S DE LA CARTE MERE

Les interfaces implantées sur la carte mère occupent les adresses d'E/S du 8086 comme suit :

ADRESSES (hex)	Usage en sortie	Usage en entrée
000-00F	Contrôleur DMA 8237	Contrôleur DMA 8237
010-01F	Ne pas utiliser	Ne pas utiliser
020-021	Contrôl. interrup. 8259	Contrôl interrup 8259
022-03F	Ne pas utiliser	Ne pas utiliser
040-042	Load Count (0-2) PIT 8253	Read Count (0-2) PIT 8253
043	Mode Load PIT 8253	Indéfini
044-05F	Ne pas utiliser	Ne pas utiliser
060	Sans effet	Code clavier Port A ou Système Status-1
061	Port B Contrôle système	Port B (relecture)
062	Sans effet	Port C Sys. Status-2
063	Sans effet	Ne pas utiliser
064	Ecriture Status-1	Ne pas utiliser
065	Ecriture Status-2	Ne pas utiliser
066	System reset	Ne pas utiliser
067-06F	Ne pas utiliser	Ne pas utiliser
070	Adresse pour RTC 146818	Ne pas utiliser
071	Donnée pour RTC 146818	Donnée RTC 146818
072-077	Ne pas utiliser	Ne pas utiliser
078	Efface coord. X souris	Coordonnée X souris
079	Ne pas utiliser	Ne pas utiliser

ADRESSES (hex)	Usage en sortie	Usage en entrée
07A	Efface coord. Y souris	Coord. Y souris
07B-07F	Ne pas utiliser	Ne pas utiliser
080	Ne pas utiliser	Ne pas utiliser
081	Reg. de page, Canal2 du DMA	Ne pas utiliser
082	Reg. de page, Canal3 du DMA	Ne pas utiliser
083	Reg. de page, C.0,1 du DMA	Ne pas utiliser
084-09F	Ne pas utiliser	Ne pas utiliser
0A0	Contrôle du masque NMI	Ne pas utiliser
0A1-0BF	Ne pas utiliser	Ne pas utiliser
0C0-0FF	Réservé	Réservé
378	Latch Printer Data	Latch Printer Data
379	Ne pas utiliser	Etat imprimante
37A	Contrôle imprimante	Contrôle impr.
37B-37F	Ne pas utiliser	Ne pas utiliser
3B0-3BF	Regitres CRTC mode Mono	Regitres CRTC mode Mono
3C0-3CF	Registres contrôleur vidéo	Registres contrôleur vidéo
3F0-3F1	Registres CRTC mode couleur	Registres CRTC mode couleur
3F2	Sélection lecteur	Ne pas utiliser
3F3	Ne pas utiliser	Ne pas utiliser
3F4	Ne pas utiliser	Etat du 765 FDC
3F5	Données du 765 FDC	Données du 765 FDC
3F6-3F7	Ne pas utiliser	Ne pas utiliser
3F8-3FF	Contrôle/Donnée du 8250 UART Tx	Contrôle/Donnée du 8250 UART Rx

Latch Printer Data : Verrou de données de l'imprimante

1.4 CANAUX D'E/S DU BUS D'EXTENSION

Les adresses d'E/S du CPU 8086 sur le bus d'extension sont les suivantes :

ADRESSES (hex)	USAGE
200-20F	Interface manette jeux externe
210-217	Unité d'extension du bus externe
220-24F	Réservé
278-27F	Port imprimante externe
2F0-2F7	Réservé
2F8-2FF	Port série asynchrone RS232C externe
300-31F	Carte prototyping externe
320-32F	Contrôleur disque dur externe
380-38F	Port série RS232C SDLC externe
3A0-3AF	Réservé
3B0-3BB	Carte Monochrome externe
3BC-3BF	Port imprimante
3C0-3CF	Carte graphique externe
3D0-3DF	Carte couleur/graphique externe

Si l'on demande une adresse d'E/S au dessus de 03FFh, on est ramené à la zone 0000h-03FFh.

Les contrôleurs des clusters externes à 0790h-0793h, 0B90h-0B93h, 1390h-1393h, 2390h-2393h sont redirigés vers les adresses d'E/S 0390h-0393h.

1.5 ACCES DIRECT A LA MEMOIRE

(DMA pour Direct Memory Access)

L'AMSTRAD PC 1640 comprend sur la carte système quatre canaux DMA en utilisant un contrôleur DMA 8237-4 et des registres de page programmables pour étendre sa zone d'adressage de 64 K à la zone d'adressage totale du processeur qui est de 1 Méga octets. Chaque canal permet le transfert de blocs de données d'une taille maximum de 64 K par page. Les canaux DMA servent aux transferts de données sur 8 bits entre les dispositifs d'E/S (8 bits) et la mémoire 8 ou 16 bits.

En mode périphérique (esclave) les lignes d'adresses d'E/S du CPU A0-A3 sont connectées conventionnellement de sorte que 16 codes de commandes apparaissent. Leur ordre est précisé dans la documentation du 8237 (voir section 3.5).

L'horloge (CLK) du contrôleur DMA est cadencée à 4 MHz (+/-0,1%). En mode maître, on ajoute 1 temps d'attente durant les transferts sur les canaux 1, 2 et 3. Il en résulte 5 temps horloge du bus DMA de 1,25 uS. Les transferts par le canal 0 ont un cycle de bus de 4 temps d'horloge de 1uS.

Les signaux du canal DMA sont les suivants :

Canal DMA	USAGE
0	Sortie OUT1 du timer 8253 pour le rafraîchissement des mémoires
1	Réservé à l'utilis. du bus d'extension Sert au port série SDLC externe
2	Sortie DRQ du contrôleur de disquettes Disponible sur le bus d'extension
3	Contrôleur disque dur. Disponible sur le bus d'extension.

1.5.1 Registres de page DMA (Page Registers)

Les canaux DMA 1, 2 et 3 peuvent adresser les 1 Mo de la zone d'adressage du CPU grâce aux registres de page qui leur sont associés. Il y a trois registres DMA, chacun correspondant à l'un des canaux. Chaque registre de page définit pour son canal la page de 64 K (parmi les 16 disponibles sur le 1 Mo) dans laquelle le transfert va avoir lieu. Les registres de page sont statiques, de sorte que les limites de page sont multiples de 64 K.

L'affectation des bits dans un registre de page est la suivante :

Bit	Utilisation en sortie
7-4	Non connectés
3	Bit d'adresse A19
2	Bit d'adresse A18
1	Bit d'adresse A17
0	Bit d'adresse A16

1.5.2 Initialisation DMA

Après un reset, la routine d'initialisation du système (ROS) configure le contrôleur DMA 8237 en affectant le mode de fonctionnement suivant au canal 0 (sur le rafraîchissement de la mémoire vive) :

Fonction	Etat initialisé
Comptage des mots	Transferts 64 K
Registre Mode	Lecture Autoinitialisation Incrémentation Mode Single
Registre Commande	Inhibe mémoire à mémoire Active le contrôleur Timing normal Priorité fixe Ecriture retardée DREQ actif haut DACK actif bas
Registre de masque	Bit du Canal 0 effacé

A la mise en route ou après un reset, les registres de page DMA sont indéfinis et initialisés à 0 par le ROS et toutes les adresses internes du 8237 pour les canaux 1 à 3 sont initialisées avec un état comparable à l'initialisation du canal 0 décrite ci-dessus.

Le PC1640 se conformant à la norme industrielle, il n'autorise pas les transferts DMA mémoire à mémoire. Des considérations de timing l'interdisent.

1.6 LES INTERRUPTIONS SYSTEME

Les interruptions NMI (Non Maskable Interrupt) du CPU et le contrôleur d'interruptions 8259A-2 autorisent 9 niveaux d'interruptions hardware. Tous les niveaux, y compris les NMI, sont masquables par logiciel.

La ligne d'adresse d'E/S A0 du CPU est connectée conventionnellement de sorte que les codes de commande apparaissent dans l'ordre décrit dans la documentation du 8259. La broche SP/EN est maintenue au niveau haut ce qui signifie que le périphérique doit voir la mémoire tampon vidée par hardware et considéré comme un maître et non un esclave.

1.6.1 Niveaux d'interruptions

Les niveaux d'interruptions sont répartis comme suit :

Niveau	Fonction affectée
NMI	Erreur de parité mémoire et sortie NDPINT du 8087
0	Sortie Out0 du timer/compteur 8253
1	Réception des codes d'exploration clavier
2	Sortie IRQ de l'horloge temps réel 146818 Disponible sur le bus d'extension
3	Réservée à l'usage du bus d'extension Utilisée par le port série externe asynchrone (secondaire) et le port série SDLC externe
4	Sortie INTRPT de l'UART 8250. Disponible sur le bus d'extension. Utilisée par le port série SDLC externe
5	Contrôleur disque dur. Disponible sur le bus d'extension.
6	Sortie INT du contrôleur de disquettes. Disponible sur le bus d'extension.
7	Port imprimante parallèle. Disponible sur le bus d'extension. Utilisée par les ports des imprimantes 2 et 3 (externes) et par le contrôleur vidéo Monochrome externe.

Le ROS initialise les bits d'adresses du 8259 de telle sorte que IRQ0 à IRQ7 prennent, pour le CPU, les vecteurs d'interruptions 8 à 15. Les NMI apparaissent au vecteur d'interruption 2 du CPU.

1.6.2 Initialisation du contrôleur d'interruptions

Après un reset, la routine d'initialisation du ROS configure le contrôleur d'interruptions 8259 de la manière suivante :

système 8086,
mode Single (c'est-à-dire solo ou non cascadé),
emboitement total normal (pas spécial),
déclenchement sur fronts,
bufférisé en mode esclave,
EOI (fin d'interruption) normale (non automatique),
priorité fixe : niveau 0 la plus haute, niveau 7 la plus basse.

Le ROS initialise les bits d'adresses du 8259 de telle sorte que IRQ0 à IRQ7 prennent pour le CPU les vecteurs d'interruptions 8 à 15. NMI est configuré pour le vecteur d'interruption 2 du CPU.

1.6.3 Contrôle de masque NMI

Le contrôle de masque NMI est un registre en écriture seule à l'adresse d'E/S 0A0H et qui permet au logiciel d'inhiber ou d'autoriser les interruptions non masquables du CPU (NMI). Les bits sont affectés selon le tableau ci-dessous :

Bit	Utilisation en sortie
Bit 7	Autorise NMI
6-0	Non connectés

A la suite d'un reset, NMI est inhibée.

NMI peut être connectée au coprocesseur arithmétique 8087, au circuit de vérification de parité de la mémoire et au bus d'extension I/OCHCK (I/O Channel Check).

1.7 LES TIMERS D'INTERVALLE PROGRAMMABLES

On dispose (aux adresses d'E/S 040h-043h) de trois timers/compteurs programmables du PIT (Programmable Interval Device) 8253.

Ils sont définis comme suit :

Compteur	Utilisation
0	Timer d'usage général
1	Utilisé par le canal DMA 0 (pour le rafraîchissement dynamique des mémoires)
2	Génération de sons par le haut-parleur

1.7.1 Configuration du timer

Les timers 8253 sont configurés ainsi :

Fonction	Configuration
CLK0,1,2	1,19318 MHz +/- 0,1% (54,925493 ms par comptage)
GATE 0,1	Toujours "ON"
GATE 2	Contrôlée via Port B (Canal de contrôle système) Sortie modulation du haut parleur.
OUT 0	Interruptions sur l'entrée PIC IR0 du 8259
OUT 1	Requêtes sur l'entrée DREQ0 du 8237 (DMA)
OUT 2	AND logique avec port B (Canal de cont. système) Sortie contrôle haut-parleur. Va aussi vers le port C (canal système Status-2) comme une entrée

1.7.2 Initialisation du compteur 1

Après un reset, la routine du ROS initialise le PIT de façon à ce que le compteur 1 (rafraîchissement dynamique des mémoires) fasse fonction de générateur de tempo en produisant un signal périodique (15,13 uS). Il n'existe pas de restrictions concernant l'initialisation et la programmation des compteurs 0 et 2.

1.8 ETAT ET CONTROLE DU SYSTEME

La carte prévoit deux canaux d'entrée d'état système et quatre canaux de sortie. Les ports A, B et C émulent un périphérique pré-programmé (PPI) 8255. Ils occupent les emplacements 060h à 06Fh de l'espace adressable des E/S. Le port B est programmé pour le contrôle en sortie, le port A est programmé soit pour l'entrée Status-1, soit pour l'entrée de données du clavier et le port B est programmé pour l'entrée Status-2.

Les ports A, B, et C émulent un PPI 8255 qui a été configuré comme suit :

Group A Mode 0,	Group B Mode 0,
Port A = entrée	Port B = sortie,
Port C(U) = entrée,	Port C(L) = entrée.

Contrairement à ce qui arrive avec un 8255, un reset ou une mise sous tension ne remettent pas en cause cette configuration.

1.8.1 Port B : Contrôle du système

Le canal réservé au contrôle du système est situé à l'adresse d'E/S 061h. Assignement des bits :

Bit (PBn)	Utilisation en sortie
7	Autorise Status-1/Inhibe code clavier sur Port A
6	Autorise l'entrée de l'horloge clavier
5	Empêche une erreur de parité d'entraîner NMI.
4	Inhibe le contrôle de parité de la RAM système on-board
3	Indéfini
2	Active octet faible du Port C/désactive octet fort (voir 1.8.3)
1	Contrôle haut-parleur
0	GATE 2 du 8253 (Modulation HP)

Quand le bit 7 (PB7) est à 1, les données Status-1 sont autorisées sur le port A, les entrées de données clavier et les interruptions clavier sont inhibées. Quand le bit 7 est à 0, les données Status-1 sont inhibées sur le port A, les entrées de données clavier et les interruptions clavier sont autorisées. Les logiciels doivent remettre PB7 à 0 sinon le clavier risque d'être inopérant.

L'interface clavier fonctionne comme ceci : chaque code de touche en entrée est bloqué ce qui entraîne une interruption clavier au niveau 1. Pendant cette interruption, le signal DATA du clavier est mis au niveau bas ; cela assure la réception du code par le clavier. Dès que l'interruption est annulée, le clavier peut envoyer un nouveau code par le signal Data.

Quand PB5 est à 1, cela interdit que la détection d'une erreur de parité externe (par exemple une condition I/OCHK sur le bus d'extension) provoque une NMI même si NMI n'est pas masquée. Une fois que NMI a été interceptée elle peut être effacée par une impulsion sur PB5 (si le device externe a arrêté son signal I/OCHK).

PB2 mis à 1 autorise la lecture sur le Port C des 4 octets de poids faible du flag de la RAM installée. Quand PB2 est à 0, l'octet haut du flag de la RAM installée est lu (voir 1.8.3).

PB1 peut être basculé pour piloter le HP avec la fréquence souhaitée. Le haut-parleur peut aussi être piloté par une forme d'onde provenant du port de sortie 2 du PIT 8253 (simultanément avec PB1).

PB0 peut être basculé pour piloter le port d'entrée du 8253 et par conséquent moduler le fonctionnement du compteur 2 ce qui permet de contrôler le haut-parleur. Toutes ces manipulations peuvent être menées simultanément pour produire des effets sonores variés.

1.8.2 Port A : entrée Status-1 / Codes clavier

Le port A est accessible en lecture seule à l'adresse d'E/S 060h. L'assignation des bits est la suivante :

Bit (PAn)	Status-1	Entrée clavier
7	Toujours 0	KBD7
6	2ième floppy installé	KBD6
5	DDM1 (Default Display Mode bit 1)	KBD5
4	DDM0	KBD4
3	Toujours 1	KBD3
2	Toujours 1	KBD2
1	NDP 8087 installé	KBD1
0	Toujours 1	KBD0

Quand le bit 7 du port B (PB7) est mis à 1, la lecture du Port A charge Status-1. Quand PB7 est à 0, lire le Port A charge les données clavier.

Le mode d'affichage par défaut (DDM1, DDM0) est mis en place par le ROS durant l'initialisation selon le tableau suivant :

DDM1	DDM0	Mode d'affichage par défaut sélectionné
0	0	IGA activé ou carte d'extension dans bus d'extension
0	1	Carte CGA installée dans bus d'extension en mode texte 40x25 caractères, blanc surbrillant sur noir.
1	0	Carte CGA installée dans bus d'extension en mode texte 80x25 caractères, blanc surbrillant sur noir.
1	1	Carte Monochrome externe 80x25 caractères

Quand IGA est activé, les switches 1-5 déterminent le mode d'affichage par défaut et si IGA est le contrôleur vidéo principal ou secondaire. Voir la section 1.22.

Quand l'IGA est inhibé, le ROS fixe le mode d'affichage par défaut en fonction des switches SW6, SW7 et SW10.

Si une carte EGA ou autre est installée, reportez-vous à son guide d'utilisation pour déterminer les modes d'affichage.

1.8.3 Port C : entrée Status 2

Port C est accessible en lecture seule à l'adresse d'E/S 062h.

L'affectation des bits est la suivante :

Bit (PCn)	Utilisation en entrée	
7	Erreur de parité de la RAM système on board	
6	Erreur de parité externe (I/OCHCK du bus d'extension)	
5	Sortie OUT2 du 8253	
4	Indéfini	
	Poids faible ou Poids fort (selon PB2)	
3	RAM3	Indéfini
2	RAM2	Indéfini
1	RAM1	Indéfini
0	RAM0	RAM4

PC7 est forcé à l'état 0 quand la vérification de parité de la RAM du système est inhibée par PB4.

Quand PB5 est haut, et donc quand une condition I/OCHCK (erreur de parité externe) ne provoque plus une NMI, PC6 reflète l'état de l'entrée I/OCHCK sinon, il reflète l'état verrouillé de I/OCHCK.

La valeur de RAM4-RAM0 est fixée en fonction de la quantité de RAM installée sur la carte mère (on-board) :

RAM4	RAM3	RAM2	RAM1	RAM0	
0	1	1	1	0	512 Koctets
0	1	1	1	1	544 Koctets
1	0	0	0	0	576 Koctets
1	0	0	0	1	608 Koctets
1	0	0	1	0	640 Koctets

Etant donné que le PC 1640 est livré avec 640 K de RAM, c'est à cette valeur qu'il faut s'attendre. Une autre valeur indique une erreur de configuration mémoire.

Voir la section 1.8.1 à propos de la configuration du Port B pour la lecture des bits de segments installés de la RAM.

1.8.4 Ecriture Système Status-1

Le registre Write System Status-1 (WSS1) est un registre à écriture seule (adresse d'E/S 064) qui est initialisé par le ROS à partir des valeurs obtenues de la NVR (Non Volatile RAM, RAM sauvegardée par piles) et les switches de configuration. WWS1 est utilisé en conjonction avec l'émulation du Port A de la PPI 8255.

L'affectation des bits de WSS1 est la suivante :

Bit	Utilisation en sortie
7	Pas d'effet
6	PA6 : Deuxième floppy installé
5	PA5 : DDM1
4	PA4 : DDM0
3-2	Pas d'effet
1	PA1 : 8087 installé
0	Pas d'effet

1.8.5 Ecriture système Status-2

Le registre Write System Status-2 (WSS2) est un registre accessible en écriture seule par l'adresse d'E/S 65h et qui est initialisé par le ROS en fonction du test de la taille mémoire faisant partie du processus d'autotest de l'initialisation de mise en marche. Il est utilisé en conjonction avec l'émulation du Port C de la PPI 8255. Les bits ont les fonctions suivantes :

Bit	Utilisation en sortie
7	PC2 (Poids fort) : Indéfini
6	PC1 (Poids fort) : Indéfini
5	PC0 (Poids fort) : Indéfini
4	PC3 (Poids fort) : RAM4
3	PC3 (Poids faible) : RAM3
2	PC2 (Poids faible) : RAM2
1	PC1 (Poids faible) : RAM1
0	PC0 (Poids faible) : RAM0

Poids fort signifie l'octet de poids fort (MSB).
Poids faible signifie l'octet de poids faible (LSB).

Etant donné que le PC 1640 possède 640 K de RAM, la valeur attendue pour PC3-PC0 est (en binaire) 10010.

1.8.6 Reset système

Tout accès en écriture à l'adresse d'E/S 066h, quel que soit la valeur écrite, amène le hardware à générer une réinitialisation du système de 512 uS et à émettre un signal sur la ligne reset du bus d'extension. Le contenu de la RAM on board est préservé après une réinitialisation.

1.9 L'HORLOGE TEMPS REEL

Une horloge temps réel (RTC pour Real Time Clock) HD146818 est installée avec sa RAM et est sauvegardée par 4 piles de type AA non rechargeables. L'horloge fournit l'heure du jour avec alarme, un calendrier sur un siècle, une interruption périodique programmable et 50 octets de RAM statique. Cette RAM est appelée Non Volatile RAM (NVR) et sert à conserver les données de configuration du système telles que le nombre de lecteurs de disquette, la taille mémoire, les paramètres des E/S série et le mode vidéo par défaut. Le ROS gère la checksum de la NVR et réinitialise les données de configuration à des valeurs prédéfinies s'il détecte une erreur checksum sur la NVR (ce qui détruit la configuration précédente). Bien qu'un accès hardware direct à la NVR soit possible, il est recommandé d'utiliser l'interruption de la fonction ROS avancée (interruption 21) pour écrire dans la NVR car seule celle-ci permet le maintien correct de la valeur de checksum.

Quand le système est hors tension, et que le 146818 est alimenté par les piles, les fonctions maintenues en activité sont l'horloge et la conservation des données de la RAM. Quand le système est sous tension, il n'utilise pas l'énergie des piles.

A l'entrée, l'oscillateur bat à 32,768 KHz et la requête d'interruption du 146818 est connectée au contrôleur d'interruptions du système 8159 au niveau 2 (qui est disponible sur le bus d'extension). L'entrée Power Sense (PS) de l'horloge est connectée à un capteur sensitif des piles. Quand le voltage des piles de sauvegarde devient trop bas, le bit VRT du registre D est mis à 1 indiquant que l'heure, la date et les données NVR ne sont plus valides. Lorsque cette condition est lue à la mise en route, le système affiche le message *"Please fit new batteries"* (remettez des piles neuves, SVP). La NVR reprend ses valeurs par défaut.

Toutes les caractéristiques décrites dans les documentations de la HD146818 sont applicables à l'exception du fait que les broches CKOUT (clock output) et SQW (square wave output) ne sont pas connectées à la carte mère.

Ecrire ou lire dans la NVR implique un processus en deux étapes pour chaque octet lu ou écrit. Le canal RTC Address (adresse d'E/S 070h) reçoit tout d'abord l'adresse de la NVR à laquelle on veut accéder. Puis on écrit ou lit la donnée dans le canal TRC Data (adresses d'E/S 71h). Cette possibilité d'écriture doit être employée avec circonspection si l'on ne veut pas perturber la configuration du système enregistrée dans la NVR.

1.10 PORT PARALLELE

Le port imprimante fournit une interface permettant de piloter les imprimantes compatibles Centronics 7 ou 8 bits. Le timing des signaux vers l'imprimante est entièrement modifiable par logiciel. Il y a un latch de contrôle en lecture/écriture pour envoyer les signaux de contrôle à l'imprimante, un canal printer status non verrouillé et un latch de données en lecture/écriture pour envoyer les données.

De plus, le latch de contrôle de l'imprimante peut être lu afin d'obtenir des informations sur le type du système et la configuration des switches.

1.10.1 Latch de données de l'imprimante (Printer Data Latch)

Le latch de données de l'imprimante est un registre accessible en lecture/écriture par l'adresse d'E/S 0378h. Il se compose ainsi :

Bits (Dn)	Utilisation en E/S	Polarité du câble
7	Data7	Haute
6	Data6	Haute
5	Data5	Haute
4	Data4	Haute
3	Data3	Haute
2	Data2	Haute
1	Data1	Haute
0	Data0	Haute

Le contenu du latch de données est indéterminé après un reset ou une mise en route.

1.10.2 Latch de contrôle de l'imprimante (Printer Control Latch)

Le latch de contrôle de l'imprimante est accessible en lecture/ écriture par l'adresse d'E/S 037Ah.

Bit	Utilisation en E/S	Reset State	Polarité du câble
7	PC 1640 SW7 [L]		
6	PC 1640 SW6 [L]		
5	OPT (1640)/1 (1512) [L]		
4	Autorise Int sur ACK	Faux	
3	Select. imprim.	Faux	Basse
2	Pas de reset imprim.	Vrai	Basse
1	Select. Auto Feed	Faux	Basse
0	Data strobe	Faux	Basse

Les bits marqués [L] sont en lecture seule.

Le bit D7 est un bit en lecture seule qui reflète l'état du switch 7 (à l'arrière de la machine) et retourne un 1 logique quand le switch est en position ON (et donc un 0 logique pour OFF).

Le bit D6 est en lecture seule et reflète l'état du switch 6 (à l'arrière de la machine) et retourne un 1 logique quand le switch est en position ON (et donc un 0 logique pour OFF).

Le bit D5 est le bit d'option (OPT) et peut fournir 3 différents types d'informations. Bien que cela ne soit pas documenté en tant que tel, le bit D5 est toujours à 1 sur un PC 1512. Sur le PC 1640, il sera toujours à 0 si immédiatement avant la lecture de 037Ah, le programme effectue une lecture d'E/S sur un canal d'E/S implémenté sur la carte mère du PC 1512 ayant la ligne d'adresse A7 à 1 (par exemple les canaux CGA). Il s'agit là d'un moyen simple pour qu'un logiciel détecte s'il tourne sur un PC 1640 ou un PC 1512.

Un PC 1512 donnera toujours 1 alors qu'un PC1640 donnera toujours 0 (après l'opération décrite).

En plus de cette possibilité d'indiquer le type de la machine sur laquelle tourne le programme, le bit OPT peut aussi reflèter l'état de SW9 ou de SW10. Pour connaître SW9, il faut effectuer une opération de lecture sur un canal d'E/S non implémenté sur la carte mère et dont les lignes d'adresses A14 et A7 soient toutes deux à l'état bas (par exemple le canal 0278h). Immédiatement après, on lit 037Ah ; OPT sera configuré comme SW9. Pour connaître l'état de SW10, on fait une opération de lecture sur un canal d'E/S non implémenté et dont A14 soit haute et A7 basse (par exemple le canal 4278h). Tout de suite après, on peut lire OPT. Les routines testant le bit OPT doivent inhiber les interruptions avant la lecture du canal initial (dummy) et celle de 037Ah afin d'interdire qu'une opération d'E/S (provoquée par une interruption) se produise entre la mise en état du bit OPT (par la lecture du canal initial) et la lecture de OPT. Pour SW9 et SW10, un 1 logique est retourné quand le switch est sur ON et un 0 logique est retourné quand le switch est sur OFF.

Quand l'interruption sur ACK est autorisée, le retour réponse de l'imprimante engendrera une interruption système au niveau 7 (qui est aussi disponible sur le bus d'extension).

Si les lignes de contrôle de l'imprimante, normalement pilotées par les bits verrouillés D0-D3, sont commandées par un dispositif externe, les données lues sur ce canal seront un OR logique entre les bits verrouillés et les bits envoyés de l'extérieur ; c'est-à-dire que si le bit de données est faux et que le bit externe est vrai, le bit lu sera vrai.

A la mise en route ou après un reset, le latch de contrôle revient à la norme décrite ci-dessus.

Notez qu'il s'agit là d'une interface imprimante universelle et que toutes les imprimantes n'utilisent pas la totalité des signaux disponibles d'où la possibilité de piloter des imprimantes non standard. En matière de timing, une imprimante standard Centronics s'attend généralement à ce que les données soient présentes 1 uS

avant l'activation du strobe et à ce qu'elles demeurent au moins 1 uS après sa désactivation. La durée du strobe (passage du signal) doit être comprise entre 1 uS et 500 uS. L'état Printer Busy peut être examiné dès la désactivation du strobe afin de savoir si l'on peut envoyer d'autres données.

1.10.3 Canal Printer Status

Le canal Printer Status est un registre (lecture seule) à l'adresse d'E/S 0379h. Il se compose ainsi :

Bit (Dn)	Utilisation en entrée	Polarité du câble
7	-Printer Busy	Haute
6	-Printer Acknowledge	Basse
5	Paper Out	Haute
4	Printer Selected	Haute
3	-Printer error	Basse
2	-LK1 actif	
1	-LK2 actif	
0	-LK3 actif	

Les signaux précédés du signe moins (-) sont inversés.

LK1-LK3 sont des liaisons d'options sur la carte mère installées en usine qui permettent au système d'exploitation en ROM (ROS) de choisir entre différentes variantes nationales de configuration. Selon leurs valeurs, les messages seront affichés en 7 langues différentes (état 0 à 6). Le huitième état (7) sert à activer le mode auto-test étendu (voir section 2.2). Etant donné que l'état des liaisons est inversé, la valeur obtenue par les 3 bits les plus faibles du port imprimante doivent faire l'objet d'un ou exclusif (XOR) avec des 1 pour donner le numéro de la langue.

LK1	LK2	LK3	Langue du ROS
OFF	OFF	OFF	Anglais
OFF	OFF	ON	Allemand
OFF	ON	OFF	Français
OFF	ON	ON	Espagnol
ON	OFF	OFF	Danois
ON	OFF	ON	Suédois
ON	ON	OFF	Italien
ON	ON	ON	Mode auto-test (anglais)

Notez qu'il s'agit là d'une interface d'usage général et que toutes les imprimantes n'utilisent pas la totalité des lignes d'état et que, de plus, elles n'affectent pas toujours la même signification aux conditions d'état.

Normalement, Printer Busy indique que l'imprimante ne peut pas recevoir de données : par exemple durant la réception de données, leur impression, quand elle est off-line ou après un code d'erreur de l'imprimante.

Printer Acknowledge, s'il est implémenté, sert normalement à l'imprimante à signaler qu'elle a reçu les données et qu'elle est prête à en recevoir de nouvelles. Notez aussi que Printer Acknowledge (ACK) peut aussi être défini pour provoquer une interruption (voir 1.10.2).

Le brochage du connecteur d'imprimante est illustré en section 1.14.

1.11 L'IGA (Internal Graphics Adapter)

Le contrôleur graphique interne (IGA), implémenté sur la carte mère, offre les modes EGA (Extended Graphics Adapter), CGA (Color Graphics Adapter) et monochrome (MDA pour Monochrome Display Adapter). De plus, on peut aussi fonctionner en mode Hercules (monochrome) et Plantronics (Color).

La mémoire vidéo (ou tampon de régénération) est une zone de 128 K s'étendant de A0000 à BFFFF dont la configuration varie en fonction du mode d'affichage adopté. L'adresse de départ du tampon vidéo (effectivement utilisé) peut être fixée soit à A0000, B0000 ou B8000 avec des tailles variant de 2 à 32 K octets (voir 1.11.7).

Avec un Extended Color Display (moniteur PC-ECD) installé, la "carte" IGA est capable d'afficher jusqu'à 64 couleurs en mode EGA avec une résolution de 640 points par 350 lignes. Les modes CGA et Plantronics offrent 16 couleurs (que ce soit avec un Extended Color Display ou un Color Display standard) avec une résolution de 640x200.

Pour l'écran monochrome PC-MD, IGA permet d'afficher le texte en noir et blanc en intensité "double" ou normale, en vidéo inversée, en clignotant ou souligné avec une résolution de 720x350 en mode texte. La résolution graphique maximum monochrome est de 640x350 ou de 720x348 en mode compatible Hercules.

Avec les systèmes couleur, lorsqu'un changement de mode initial est effectué via la ROM BIOS, un ensemble de 16 registres (couleurs de la palette) reçoit les valeurs standard de sorte que l'on dispose d'une sélection de 16 couleurs correspondant aux 16 couleurs diponibles à l'écran. Les champs d'attributs de couleur (que nous détaillerons plus tard) et les sélections de plans de couleur se font en fonction de la palette des 16 couleurs standard. En voici le format :

Intensité	Rouge	Vert	Bleu	Couleur
0	0	0	0	Noir
0	0	0	1	Bleu
0	0	1	0	Vert
0	0	1	1	Cyan
0	1	0	0	Rouge
0	1	0	1	Magenta
0	1	1	0	Marron
0	1	1	1	Blanc
1	0	0	0	Gris
1	0	0	1	Bleu clair
1	0	1	0	Vert clair
1	0	1	1	Cyan clair
1	1	0	0	Rouge clair
1	1	0	1	Magenta clair
1	1	1	0	Jaune
1	1	1	1	Blanc intense

Les couleurs "claires" sont obtenues par dé-saturation avec un apport de gris (blanc faible).

Notez que l'écran Extended Color Display (PC-ECD) supporte 64 couleurs avec un câble à 6 fils rgbRGB (où les lettres majuscules représentent les couleurs primaires avec une intensité de 2/3 et les lettres minuscules des signaux d'une intensité de 1/3). Etant donné que chacun des 16 registres de palette est un registre à 6 bits représentant rgbRGB (respectivement du bit de poids fort au bit de poids faible), seules 16 des 64 couleurs peuvent être affichées simultanément.

1.11.1 Affichage couleur en mode texte

En mode couleur, on dispose de deux modes texte : soit 40 caractères par 25 lignes (moyenne résolution), soit 80 caractères par 25 lignes (haute résolution). La taille mémoire (RAM vidéo) occupée par ces deux types d'écran est respectivement de 2 et 4 K. Le

tampon de régénération pour ces modes s'étend de B8000h à BFFFFh et la ROM bios d'affichage permet de disposer de 8 ou 4 (selon le mode 40 ou 80 colonnes) pages d'écran.

Les jeux de caractères sont formés à partir d'une matrice chargée en RAM et pour les systèmes possédant une résolution de 200 lignes, chacun des 256 caractères est défini en 8x8 pixels.

Lorsqu'on dispose d'un Extended Color Display (PC-ECD) et qu'on l'utilise en résolution maximum (350 lignes), les modes texte 40 et 80 colonnes sont toujours acceptés mais le générateur de caractères est programmé différemment : les caractères sont formés d'une matrice 8x14 pixels. La disposition mémoire du tampon de régénération, son origine et sa taille sont les mêmes qu'en mode 200 lignes.

L'adresse de départ dans la RAM d'affichage (RAM vidéo) est programmée via le CRTC (Cathode Ray Tube Controller). Le départ de cette RAM est toujours sur une adresse paire et correspond à la première (la plus à gauche) position de caractère de la ligne supérieure de l'affichage. Le CRTC Starting Address est un registre 16 bits qui spécifie en deux paires d'octets l'offset (le déplacement) depuis l'origine du mode d'affichage. Cela signifie que si l'on change la valeur du registre d'adresse de départ du CRTC d'une unité, ce sera la prochaine adresse de même parité dans la RAM vidéo qui constituera l'origine du tampon de régénération.

Pour afficher un seul caractère, il faut deux octets de RAM vidéo et dans chaque paire de ces octets, l'adresse paire représente le code du caractère et l'adresse impaire son attribut.

Les caractères qui se suivent sont affichés de la gauche vers la droite de la ligne. Quand la limite droite de la ligne est atteinte, la prochaine paire d'octets de la RAM vidéo viendra s'afficher à la première position (à gauche) de la ligne suivante (en dessous). L'annexe 8 donne les 256 codes de caractères et leur représentation par défaut. Notez que le bios en ROM du mode EGA permet la reprogrammation du générateur de caractères afin que l'utilisateur puisse disposer de ses propres matrices.

L'octet d'attribut permet de choisir entre 16 couleurs d'avant-plan et 8 couleurs de fond par caractère plus le clignotement, ou alors entre 16 couleurs à la fois pour le fond et l'avant-plan sans clignotement. En mode CGA, on peut programmer la couleur du bord d'écran (choisie parmi les 16 couleurs courantes).

L'octet d'attribut de chaque caractère est défini comme suit :

Bit(ATn)	Définition
7	Intensité ou autorise clignotement (fond)
6	Rouge (fond)
5	Vert (fond)
4	Bleu (fond)
3	Intensité (avant-plan) ou sélection Map A/B
2	Rouge (avant-plan)
1	Vert (avant-plan)
0	Bleu (avant-plan)

Le bit 7, Intensity or Enable Blink, change de fonction selon la valeur du registre de contrôle de mode (Mode Control Register). En mode EGA, le bit 3 du Mode Control Register (adresse d'E/S 03C0h indice 10) permet de choisir entre le contrôle de l'intensité ou du clignotement dans les modes d'affichage alphanumériques standard. En mode compatible CGA, le bit 5 du registre Mode Control (adresse d'E/S 03D8h) assure la sélection entre l'intensité et le clignotement.

En mode EGA, le bit 3 (celui de l'intensité de l'avant-plan) peut servir comme bit de sélection de "Character Map" afin de disposer de 512 caractères affichables simultanément à l'écran. L'option de Character map étendue est traitée dans la section 1.11.8 (Registres des modes graphiques IGA étendus).

Remarquez qu'en mode EGA, la disposition des bits IRGB sélectionne en fait l'un des 16 registres de palette et les signaux de la couleur (câble étendu à 6 fils) sont produits selon le contenu du registre de palette sélectionné.

Si l'on n'a pas modifié les registres de palette depuis le changement de mode, les couleurs disponibles seront les 16 couleurs standard listées au début de cette section. La table ci-dessous donne les valeurs à affecter aux registres de palette afin d'obtenir la palette standard de 16 couleurs :

Reg.	rgb	RGB	Couleur
0	000	000	Noir
1	000	001	Bleu
2	000	010	Vert
3	000	011	Cyan
4	000	100	Rouge
5	000	101	Magenta
6	000	110	Marron
7	000	111	Blanc (Gris1)
8	111	000	Gris (Gris2)
9	111	001	Bleu clair
A	111	010	Vert clair
B	111	011	Cyan clair
C	111	100	Rouge clair
D	111	101	Magenta clair
E	111	110	Jaune (clair)
F	111	111	Blanc intense

Les lettres minuscules représentent les couleurs secondaires avec une intensité de 1/3 chacune et les lettres majuscules représentent les couleurs primaires avec une intensité de 2/3 chacune.

1.11.2 Affichage texte monochrome

En monochrome, on ne dispose que d'un mode d'affichage texte : 80 caractères par 25 lignes, soit une résolution de 720 points par 350 lignes. Les tampons de régénération de l'affichage prennent place entre B0000h et B7000h de la RAM vidéo et occupent 4 K chacun. Le bios en ROM de l'affichage permet de définir 8 pages d'écran.

Le tampon 0 se trouve en B0000-B0F9F, le tampon 1 en B1000-B1F9F et les 6 autres s'étalent jusqu'à B7000 en commençant à une adresse multiple de 1000h.

En mode EGA, le jeu de caractères est formé par une matrice chargée en RAM et chaque caractère est défini par une matrice de 9x14 pixels.

L'adresse de départ du CRTC se programme de la même manière qu'en couleur et la paire de 2 octets (caractère et attribut) est disposée dans la RAM comme elle l'est en mode couleur. Cependant l'octet d'attribut assure des fonctions différentes étant donné qu'il n'y a pas de signaux IRBG à envoyer au moniteur. Video et Intensity subsistent tout de même. L'octet d'attribut monochrome est composé ainsi :

Fonction	**Bits**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Effacé	Fd	I/C	0	0	0	I	0	0	0
Souligné	Fd	I/C	0	0	0	I	0	0	1
Normal	Fd	I/C	0	0	0	I	1	1	1
Inverse	Fd	I/C	1	1	1	I	0	0	0

Le bit 7, Intensité du fond/Clignotement (en anglais B/I Pour Background Intensity / Blink Enable) change de fonction en relation avec le registre Mode Control. En mode EGA, le bit 3 du registre Mode Control (adresses d'E/S 03C0h, indice 10) effectue la sélection entre fond intense quand le mode est inversé ou en clignotement. Dans le mode compatible MDA, le bit 5 du Registre Mode Control (adresse d'E/S 03B8h) permet de choisir entre Intensité ou clignotement. En mode texte, le mode compatible Hercules fonctionne de la même manière.

Le bit 3 est le bit d'intensité de l'avant-plan et contrôle donc l'intensité quand on ne se trouve pas en vidéo inverse ou effacée. Il est aussi nécessaire de faire baisser le contraste de l'écran PC-MD pour pouvoir observer les différences de niveaux vidéo.

1.11.3 Affichage graphique couleur

L'écran PC-CD (Standard Color Display) permet d'afficher 200 lignes horizontales et offre un choix de 2 résolutions graphiques : soit 320 pixels par ligne avec 4 couleurs par pixel, soit 640 pixels par ligne en 2 couleurs. Ce sont les modes compatibles CGA. Les modes supplémentaires 16 couleurs (EGA) sont aussi accessibles avec ces deux résolutions. En modes compatibles CGA, le tampon de régénération du 320x200-4 couleurs et du 640x200-2 couleurs commence à B8000 et occupe 16 K par page. Grâce au bios en ROM de l'IGA, on peut avoir respectivement 8 ou 4 pages écran selon que l'on est en 320 ou 640 pixels.

Pour les modes EGA 16 couleurs, le tampon commence à A0000 et est divisé en 4 plans de 64 K correspondant aux bits de Bleu, Vert, Rouge et Intensité. Chaque plan peut être manipulé individuellement par le CPU qui peut, de plus, sélectionner deux plans pour y écrire simultanément la même donnée. Chaque bit de données individuel peut être au choix activé ou inhibé durant l'opération d'écriture du CPU dans la page graphique. La section consacrée aux registres (1.11.8 ci-dessous) explique les méthodes disponibles pour le contrôle de l'affichage graphique.

Le PC-ECD (Extended Color Display) supporte les modes 200 lignes décrits ci-dessus en mode 640x350 avec 16 ou 64 couleurs. Pour ce mode haute résolution, le tampon commence à A0000 et demande 28 K de RAM par page. On dispose de 2 pages d'écran en mode 350 lignes. La sélection des 16 couleurs se fait avec la méthode des 4 plans décrite précédemment. Le choix de 16 couleurs parmi les 64 possibles se fait en affectant de nouvelles valeurs au registres de palette (voir la description du mode texte couleur).

1.11.4 Mode graphique basse résolution (320x200)

En mode graphique basse résolution, chaque ligne de pixels occupe 80 octets de mémoire. Chaque pixel nécessite 2 bits de sorte qu'un octet vaut 4 pixels. Le pixel le plus à gauche est défini par les 2 bits de poids fort de l'octet et les 3 autres pixels occupent chacun une paire de bits de gauche à droite.

Ce champ de 2 bits pour chaque pixel spécifie l'une des 4 couleurs et peut appartenir à l'une des trois palettes ci-dessous :

Couleur	Palette 0	Palette 1	Palette 2
0	Fond	Fond	Fond
1	Vert	Cyan	Cyan
2	Rouge	Magenta	Rouge
3	Jaune	Blanc	Blanc

Le tampon de régénération vidéo du mode basse résolution présente une occupation mémoire discontinue avec les 100 lignes de pixels paires (0, 2, 4, ...198) occupant la mémoire graphique de B8000 à B9F3F et les 100 lignes impaires (1, 3, 5, ...199) allant de BA000 à BBF3F. La carte mémoire est la suivante :

320 pixels (2 bits)

B8000-	ligne de pixels 0 (80 octets)	-B804F
B8050-	ligne de pixels 2	-B809F
B80A0-	ligne de pixels 4	-B80EF
	.	
	.	
B9EF0-	ligne de pixels 198	-B9F3F
BA000-	ligne de pixels 1	-BA04F
BA050-	ligne de pixels 3	-BA09F
BA0A0-	ligne de pixels 5	-BA0EF
	.	
	.	
BBEF0-	ligne de pixels 199	-BBF3F

La "cartographie" d'un octet de RAM graphique en mode basse résolution est :

Bit RAM	7 : 6	5 : 4	3 : 2	1 : 0
Pixel	0	1	2	3
Bit du pixel	1 : 0	1 : 0	1 : 0	1 : 0

1.11.5 Mode graphique moyenne résolution (640x200)

En mode graphique moyenne résolution, chaque ligne de 640 pixels se compose de 80 octets. Chaque pixel est défini par un bit de sorte qu'un octet représente 8 pixels. Le pixel le plus à gauche correspond au bit de poids fort de l'octet et les autres suivent de gauche à droite. En moyenne résolution, les deux couleurs sont soit noir (bit du pixel à 0) soit l'une des 16 couleurs sélectionnée par le registre d'avant-plan de la palette si le bit est à 1.

Un octet de RAM graphique en moyenne résolution se présente comme ceci :

Bit de la RAM	7	6	5	4	3	2	1	0
Pixel	0	1	2	3	4	5	6	7

L'implantation mémoire d'une ligne de pixels dans la RAM vidéo est sur le même modèle de configuration discontinue que celle de la basse résolution : toutes les lignes paires de B8000 à B9F3F suivies par toutes les lignes impaires de BA000 à BBF3F.

1.11.6 Mode graphique haute résolution (640x350)

En mode haute résolution, comme en moyenne résolution, chaque ligne de pixels occupe 80 octets de mémoire. La différence principale tient à ce que l'on trouve 350 lignes qui sont définies par un bloc de mémoire continu allant de A0000 à AFFFF. Le BIOS en ROM IGA accepte deux pages d'écran par reprogrammation du registre d'adresse de départ du CRTC. La page 0 va de A0000 à A6D5F et la page 1 de A8000 à AED5F.

La cartographie mémoire du mode haute résolution est très simple : on trouve 350 blocs de 80 octets contigus. La correspondance entre bit et pixel est la même qu'en moyenne résolution.

Cette disposition s'applique à la fois au mode EGA monochrome (mode BIOS 15) et au mode EGA haute résolution couleur (mode BIOS 16). L'implantation mémoire du mode graphique Hercules est passablement plus complexe et correspond à une carte (au sens "cartographie" et non pas carte d'extension) segmentée en 4 morceaux. Ce sujet sera traité dans la section 1.11.12 avec les registres de contrôle du mode Hercules.

1.11.7 Les modes du BIOS IGA

Le BIOS IGA peut configurer le hardware de manière à obtenir les 12 modes d'affichage différents accessibles au PC 1640. Le tableau suivant donne les modes autorisés par la ROM BIOS. Les numéros 8 à 12 correspondent à des modes d'autres adaptateurs graphiques et ne sont donc pas accessibles avec la ROM BIOS IGA.

Mode BIOS	0	1	2	3	4	5	6	7	13	14	15	16
Type	Texte	Texte	Texte	Texte	Graph	Graph	Graph	Texte	Graph	Graph	Graph	Graph
Colonnes	40	40	80	80	320	320	640	80	320	640	640	640
Lignes	25	25	25	25	200	200	200	25	200	200	350	350
Couleurs	NB	16	NB	16	4	NB	2	Mono	16	16	Mono	16/64
Taille car.	8x8	8x8	8x8	8x8	8x8	8x8	8x8	9x14	8x8	8x8	9x14	9x14
Origine tampon	B8000	B8000	B8000	B8000	B8000	B8000	B8000	B0000	A0000	A0000	A0000	A0000
Taille tampon	32768	32768	32768	32768	32768	32768	32768	32768	65536	65536	65536	65536
Taille page	2048	2048	4096	4096	16384	16384	16384	4096	8192	16384	32768	32768
Nombre pages	16	16	8	8	2	2	2	8	8	4	2	2

L'origine du tampon de régénération est indiquée en hexadécimal étant donné qu'il s'agit d'une adresse. Toutes les autres valeurs sont en décimal afin de donner un ordre de grandeur.

Le BIOS IGA permet l'utilisation de plusieurs pages d'écran virtuel et peut être appelé pour activer la page de son choix. La page active par défaut est toujours la page 0 et elle commence à l'adresse d'origine du tampon. Les pages successives sont implantées vers le haut en incrémentant l'adresse d'origine de la taille de la page telle qu'elle apparaît ici. L'équation permettant d'obtenir l'origine de la page est la suivante : Origine page = Origine tampon + (Numéro de page * Taille de la page). Les pages sont numérotées de 0 à n-1 où n est le nombre de pages disponibles.

On remarque que le nombre maximum de pages des modes 0 et 1 est noté ici comme étant 16. Bien que cela soit vrai d'un point de vue matériel, le BIOS IGA ne manipule que les 8 premières pages. Ses routines de génération de caractères ne fonctionnent pas correctement sur les pages 8 à 15. Ces pages sont cependant accessibles par une manipulation directe de la mémoire.

1.11.8 Les registres de contrôle IGA

Les registres de contrôle de l'IGA sont assez complexes mais permettent une grande variété de modes d'affichage texte et graphiques. La ROM Operating System du PC 1640 (ROS) et la ROM BIOS IGA permettent un interfaçage plus simple avec le matériel et l'on devra sans doute les préférer pour le développement de programmes avec un certain souci de portabilité vers du matériel futur qui pourrait ne pas être compatible au niveau hardware.

Registres IGA spéciaux

Trois registres spéciaux contrôlent le mode opérationnel de l'IGA et les caractéristiques de manipulation de ces modes. La ROM BIOS IGA et l'utilitaire DISPLAY fourni avec le PC 1640 se chargent généralement de la programmation de ces registres. L'IGA ne fait pas d'auto-configuration, c'est-à-dire qu'il restera dans le mode d'émulation hardware choisi jusqu'à ce qu'un programme comme DISPLAY manipule les registres de contrôle. D'autre part, certaines applications possédant des drivers pour des matériels donnés se serviront de ces registres pour leurs besoins particuliers. Les registres spéciaux de l'IGA sont :

Nom	Adresse du Port
IGA Extended Mode Control	3DB/3BB
Hercules Control	3BF
Plantronics Control	3DD

Registre Extended Mode Control

Le registre Extended Mode Control contrôle le mode opérationnel général de l'IGA de façon à pouvoir se trouver en mode d'émulation EGA, CGA, MDA, Hercules ou Plantronics.

Le registre Extended Mode Control est un registre 8 bits en écriture seule accessible par les adresses d'E/S 3DB/3BB et on ne peut y écrire qu'après deux lectures successives d'E/S des adresses 3D8/3B8. Il s'agit là d'une mesure de protection contre une modification accidentelle du mode. Pour savoir quelles adresses d'E/S utiliser entre 3B8/3BB et 3D8/3DB (adresses Mono ou Couleur), il est nécessaire de savoir si le système est configuré en monochrome ou en couleur. La méthode standard est d'utiliser le pointeur des adresses d'E/S contrôleur vidéo du ROS situé en RAM à 00463. Cette valeur sur 16 bits sera soit 3B4, soit 3D4 selon que (respectivement) le système est monochrome ou couleur.

Le format du registre Extended Mode Control est le suivant :

Bit	Fonction en sortie
7	Polarité Vsync, effacement du bord
6	Active les modes spéciaux
5	Inhibe la palette et les registes Overscan
4	Verrouille les registres de timing du CRTC
3	Active les jeux de caractères alternés sur plan 3
2	Inhibe l'effacement
1	Active le mode 132 colonnes
0	Active les modes Color Simulation

Quand le bit 7 est à 1, il force la polarité négative de la synchro verticale et efface le bord de l'écran.

Quand il est à 1, le bit 6 autorise les modes compatibles 6845, CGA, Monochrome, Hercules et PLantronics.

Mis à 1, le bit 4 empêche la modification des registres de contrôle du contrôleur vidéo qui déterminent le timing du signal de synchro verticale.

Le bit 3 (à 1) permet d'utiliser un jeu de caractères optionnel de 8 K pour le plan 3. Voir plus loin.

Le bit 2 mis à 1 désactive l'effacement d'écran en mode texte couleur CGA.

Quand il est à 1, le bit 1 active le mode 132 caractères.

Le bit 0 mis à 1 autorise les modes de simulation de couleurs.

Registre Hercules Mode

Le registre Hercules Mode est actif quand on se trouve en mode Hercules graphique monochrome. Il est accessible en écriture seule à l'adresse d'E/S 3BF. Il contrôle la configuration de la mémoire graphique et protège d'un changement accidentel des bits de mode dans le port de contrôle du mode en 3B8.

Bit	Fonction
7-2	Inutilisés
1	Autorise bit 7 du Mode Register (bit Full/Half Graphics)
0	Autorise bit 1 du Mode Register (bit Texte/Graphique)

Etant donné que le mode Hercules ressemble au mode MDA mais avec des extensions graphiques, certaines applications conçues pour le MDA pourraient accidentellement activer les bits du mode Hercules. Ce registre prévient un tel incident en forçant à 0 les bits respectifs de telle sorte que le mode texte est maintenu et que le tampon d'affichage reste à B0000.

Registre Plantronics Mode

Le mode Plantronics est un mode qui permet des couleurs de style CGA mais fournit en plus un mode 4 couleurs en 640x200 et 16 couleurs en 320x200 grâce à un plan de couleur supplémentaire en BC000. Le registre Plantronics Mode (PLR) est en écriture seule à l'adresse d'E/S 3DD. Son format est le suivant :

Bit	Fonction
7	Inutilisé
6	Position du plan de couleur 0/1
5	Active la palette étendue 1
6	Active la palette étendue 2
3-0	Inutilisés

Avec le bit 6 mis à 0, le plan de couleur 0 commence en B8000 et le plan 1 en BC000. Si le bit 6 est à 1, les deux adresses de départ des plans sont échangées : plan 0 en BC000 et plan 1 en B8000. Chaque plan de couleur mesure 16K.

Quand les bits 4 et 5 sont tous les deux à 0, le mode standard CGA est activé. Quand ils sont tous les deux à 1, le bit 4 prend le dessus sur le bit 5 mais cette situation n'est pas recommandée ; elle risque d'engendrer des effets de couleurs indésirables. Le bit 6 n'a d'effet que si l'un des bits 4 ou 5 est à 1.

Registres des modes compatibles EGA

Il y a soixante cinq registres de modes compatibles EGA qui contrôlent les caractéristiques de l'environnement graphique étendu. Ces registres peuvent être regroupés en cinq groupes logiques EGC Control (externe), Attribute Controller, Sequencer, Graphics Controller et CRT Controller (CRTC). Ils figurent ici selon ce groupement :

Nom du registre	Port	L/E	Indice
EGC Control	3C2	E	-
EGC Status	3C2	L	-
Attribute Controller	3C0	E	-
Palette (0-F)	3C0	E	00-0F
Mode Control	3C0	E	10

Nom du registre	Port	L/E	Indice
Extended Graphics Border	3C0	E	11
Color Plane Enable	3C0	E	12
Horizontal Panning	3C0	E	13
Sequencer Address	3C4	E	-
Clock Mode	3C5	E	01
Color Plane Write	3C5	E	02
Character Set Select	3C5	E	03
Memory Mode Select	3C5	E	04
Graphics Controller Address	3CE	E	-
Set/Reset	3CF	E	00
Enable Set/Reset	3CF	E	01
Color Plane Compare	3CF	E	02
Data Rotate	3CF	E	03
Color Plane Read	3CF	E	04
Graphics Mode 1	3CF	E	05
Graphics Mode 2	3CF	E	06
Color No Care	3CF	E	07
Write Mask	3CF	E	08
CRTC Address	3B4/3D4	E	-
Horizontal Total	3B5/3D5	E	00
Horizontal Display End	3B5/3D5	E	01
Start Horizontal Blanking	3B5/3D5	E	02
End Horizontal Blanking	3B5/3D5	E	03
Start Horizontal Retrace	3B5/3D5	E	04
End Horizontal Retrace	3B5/3D5	E	05
Vertivcal Total	3B5/3D5	E	06
CTRC Overflow	3B5/3D5	E	07
Preset Row Scan	3B5/3D5	E	08
Maximum Scan Line	3B5/3D5	E	09
Cursor Start	3B5/3D5	E	0A
Cursor End	3B5/3D5	E	0B
Start Address High	3B5/3D5	L/E	0C
Start Address Low	3B5/3D5	L/E	0D

Nom du registre	Port	L/E	Indice
Cursor Location High	3B5/3D5	L/E	0E
Cursor Location Low	3B5/3D5	L/E	0F
Start Vertical Retrace	3B5/3D5	E	10
Light Pen High	3B5/3D5	L	10
End Vertical Retrace	3B5/3D5	E	11
Light Pen Low	3B5/3D5	L	11
Vertical Display Enable End	3B5/3D5	E	12
Offset	3B5/3D5	E	13
Underline Location	3B5/3D5	E	14
Start Vertical Blanking	3B5/3D5	E	15
End Vertical Blanking	3B5/3D5	E	16
CRTC Mode Control	3B5/3D5	E	17
Line Compare Register	3B5/3D5	E	18
Status Port	3BA/3DA	L	-

L signifie Lecture seule, E Ecriture seule et L/E lecture-écriture.

Les registres CRTC résideront soit dans la zone 3B-... pour l'affichage monochrome, soit dans la zone 3D-... pour l'affichage couleur. Cette sélection d'adresses variables se fait par le bit 0 du registre EGC Control à l'adresse d'E/S 3C2.

a. Registres EGC External Control

Les registres EGC External Control sont au nombre de deux. Ils permettent la lecture d'informations telles que la configuration de switches et permettent un certain nombre d'aménagements hardware.

Registre EGC Control (3C2 Out)

Le registre EGC Control est accessible en écriture seule par l'adresse d'E/S 3C2. Son format est le suivant :

Bit	Fonction
7	Polarité VSYNC
6	Polarité HSYNC
6	Sélection de la page texte alternée (64 K)
4	Vidéo externe active
3-2	Sélection fréquence horloge/sélection de l'inversion des switches
1	Autorisation d'accès à la RAM vidéo
0	Sélection d'adresses CRTC 3Bx/3Dx

Bit 7=0 pour une polarité VSYNC positive et bit 7=1 pour une polarité VSYNC négative.

Bit 6=0 pour une polarité HSYNC positive et bit 6=1 pour une polarité HSYNC négative.

Le bit 5 permet la sélection d'une RAM vidéo secondaire de 64 K pour les modes texte (modes BIOS 0-3 et 7). Bit 5=0 pour la page Basse (par défaut) et bit 5=1 pour la page Haute (secondaire).

Quand le bit 4 est à 1, il inhibe la vidéo interne et ouvre un canal de données pour une carte vidéo externe. Cependant, étant donné qu'aucun périphérique de ce type n'est fourni avec le PC 1640, ce bit doit toujours être écrit à 0.

Les bits 2 et 3 forment un champ de 2 bits permettant de sélectionner l'une des deux fréquences d'horloge. Quand ces deux bits sont à 0 (bit 3=0, bit 2=0), l'horloge 14 MHz sert de fréquence de base. Quand bit 3 est à 0 et bit 2 à 1, la fréquence d'horloge de base de l'IGA est 16 Mhz. Les deux autres combinaisons sont invalides pour le PC 1640. En plus de cette sélection de fréquence, les bits 2 et 3 forment un champ permettant de lire les switches 1 à 4 de la carte mère (en complément à 1) en conjonction avec le bit 4 de EGC Status.

Si le bit 1=0, le CPU n'a plus accès à la RAM vidéo et si le bit 1=1, le CPU a accès à la RAM vidéo.

Le bit 0 détermine la configuration de l'adressage du CRTC pour les adresses d'E/S en mode monochrome ou couleur. Quand ce bit est à 0, le CRTC réside à 3B4 et 3B5 pour le mode monochrome et quand le bit 0=1 le CRTC réside à 3D4 et 3D5 pour le mode couleur. Le registre Status Port est aussi "translaté" de l'adresse d'E/S 3BA en monochrome à l'adresse d'E/S 3DA en couleur.

Registre EGC Status (3C2 In)

Le registre EGC Status est un registre en lecture seule qui réside à l'adresse d'E/S 3C2.

Bit	Fonction
7	Interruption VSYNC active
6-5	Inutilisés
4	Switches
3-0	Inutilisés

Le bit 7=1 quand VSYNC est en attente et le bit 7=0 quand la requête d'interruption VSYNC a été annulée ou est inactive.

Le bit 4 fonctionne en conjonction avec les bits 2 et 3 du registre EGC Control pour former un sélecteur des switches. Quand le bit 4 est à 0, le switche sélectionné (1-4) est fermé. Le tableau suivant donne la configuration de sélection des switches :

Reg. EGC Control Out(3C2)		Reg. EGC Status In(3C2)
Bits 3	2	Bit 4
1	1	-Switch 1
1	0	-Switch 2
0	1	-Switch 3
0	0	-Switch 4

La ROM BIOS IGA lit ces switches durant la phase d'autotest de l'initialisation et affecte la valeur (en complément à 1) des switches à l'adresse système 0:448. Les bits 0-3 correspondent respectivement au complément à 1 des switches 1-4. Les 4 bits de poids fort de l'adresse 0:488 seront toujours à 1 sur un PC1640.

b. Registres du contrôleur d'attribut (Attribute Controller Register)

Le contrôleur d'attribut est un élément hardware de l'Extended Graphics Controller qui est chargé de la gestion des couleurs. Les composantes principales de cet élément logique sont les registres de palette qui permettent la sélection de 16 couleurs simultanées parmi 64.

Registre Attribute Controller Address

Ce registre d'adressage du contrôleur d'attribut est accessible en écriture seule à l'adresse d'E/S 3C0h. Lorsqu'on y écrit, sa valeur représente un indice dans l'ensemble des registres du contrôleur d'attribut qui réside aussi en 3C0 (E/S). Le processeur doit d'abord envoyer la valeur d'indice puis la faire suivre immédiatement de la valeur à affecter au registre visé. Le non-respect de cette séquence d'adressage entraînera une confusion entre les indices et les valeurs pour les opérations suivantes.

Registres de palette (00h-0Fh)

Les seize registres de palette résident aux indices 00h-0Fh du contrôleur d'attributs et chacun d'eux se voit affecter la fonction de bit suivante :

Bit	Fonction	Broche vidéo
7-6	Inutilisés	-
5	Vidéo Rouge secondaire	2(r)
4	Vert secondaire/Intensité	6(g/I)
3	Bleu secondaire/Vidéo Mono	7(b/V)
2	Rouge primaire	3(R)
1	Vert primaire	4(G)
0	Bleu primaire	5(B)

Les registres de palette ont chacun six bits qui sont arrangés de telle manière que les signaux couleurs émis correspondront (du bit de poids fort au bit de poids faible) à l'ordre rgbRGB. Le point principal à noter est que le vert secondaire et l'intensité sont sur la même broche en sortie. Initialement, le BIOS IGA chargera les registres de palette de manière à ce que le bit de vert secondaire (bit 4) soit mis à 1 pour tous les registres d'ordre supérieur qui ont besoin que Intensity soit actif pour les affichages 16 couleurs IRGB. Ce schéma permet à la sélection de couleurs de se faire à la manière de celle du mode CGA. Chacun des champs de couleur qui comprend 4 bits (c'est-à-dire les octets d'attributs et les plans de couleurs) sélectionnera en fait son registre de palette pour le choix des signaux couleurs à émettre.

Registre Mode Control (10h)

Le Mode Control Register est un registre à écriture seule qui réside à l'indice 10h du contrôleur d'attribut (adresse d'E/S 3C0). Ce registre spécifie le mode opérationnel principal et d'autres caractéristiques des modes sélectionnés. Le registre Mode Control présente la disposition suivante :

Bit	Fonction
7-4	Inutilisés
3	Intensité du fond/Active le clignotement
2	Active le mode Line Graphics
1	Type d'affichage
0	Graphique/Texte

En mode texte (alphanumérique), le bit 3 en conjonction avec le bit de poids fort de l'octet d'attribut du texte détermine si le bit d'attribut de poids fort sera le bit d'activation du clignotement (si bit 3 est à 1) ou si le bit de poids fort d'attribut sera le bit d'intensité du fond (si le bit 3 est à 0). La fréquence de clignotement est définie ainsi : 16 cycles d'écran allumés, 16 cycles d'écran éteints ce qui représente la moitié de la fréquence de clignotement du curseur. En mode couleur graphique, le bit 3 détermine si un pixel particulier va passer alternativement d'une couleur à l'autre par une bascule de COL3 à la fréquence de clignotement.

Le bit 2 active le jeu de caractères graphiques spéciaux initialement conçu pour le MDA en faisant en sorte que le neuvième point soit le même que le huitième point du caractère défini par une matrice 9x14.

Lorsque le bit 1 est vrai, il sélectionne les attributs d'affichage monochrome et lorsqu'il est à 0, il sélectionne les attributs d'affichage couleur.

Le bit 0 mis à 1 sélectionne le mode graphique et mis à 0, il sélectionne un mode texte.

Registre Extended Graphics Border (11h)

L'Extended Graphics Border Register est accessible par l'indice 11h des registres du contrôleur d'attributs à l'adresse d'E/S 3C0h. On l'appelle aussi le registre d'Overscan. Il dispose de 6 bits et

n'est accessible qu'en écriture. Sa fonction est de spécifier la couleur du bord des modes graphiques étendus. Il peut être verrouillé en mettant à 1 le bit 5 du registre IGA Extended Mode Control.

Le format de ce registre est le même que celui des registres de palette. Lorsqu'on se trouve en mode monochrome, l'Extended Graphics Border doit être mis à 0.

Registre Color Plane Enable (12h)

Le Color Plane Enable est un registre en écriture seule accessible par l'indice 12h du contrôleur d'attribut à l'adresse d'E/S 3C0. Il sert à déterminer les plans de couleur actifs pour l'affichage et peut donc être considéré comme un sélecteur global de palette. Son format est le suivant :

Bit	Fonction
7-6	Inutilisés
5-4	Video Status Multiplex (MUX)
3	Activation du plan de couleur 3
2	Activation du plan de couleur 2
1	Activation du plan de couleur 1
0	Activation du plan de couleur 0

Quand les bits 0-3 sont mis à 1, le plan de couleur correspondant est activé.

Les deux bits 4 et 5 permettent d'exécuter un auto-test du hardware en sélectionnant les deux sorties des plans de couleur qui seront renvoyées vers les bits 4 et 5 du registre Status Port à 3BA/3DA. Le tableau suivant donne la correspondance entre les bits d'état vidéo MUX et le STATUS PORT :

Video Status MUX (3C0) Bits 5	4	Reg. Status Port (3XA) Bits 5	4
0	0	R	B
0	1	r	g
1	0	b	G

La lettre G ou g représente Vert ou vert.

Les lettres majuscules représentent les couleurs primaires et les lettres minuscules les couleurs secondaires. Le cas 1, 1 n'est pas utilisé.

Registre Horizontal Panning (13h)

Il s'agit d'un registre en écriture seule accessible à l'indice 13h du contrôleur d'attribut à l'adresse d'E/S 3C0. Il permet de choisir le nombre de pixels sur lequel on fait glisser l'affichage vers la gauche (horizontalement). Le panning est accessible en mode graphique ou en mode texte. L'amplitude du décalage en mode texte monochrome est au maximum de 9 pixels. Pour les autres modes, il est de 8 pixels. Les bits du registre Horizontal Panning reçoivent l'affectation suivante :

Bit	Fonction
7-4	Inutilisés
3-0	Valeur du panning horizontal.

Pour le mode monochrome texte, la séquence du glissement sur 9 pixels est la suivante : tout d'abord le pixel 8, puis les pixels 0 à 7. Pour tous les autres modes, le décalage se fait du pixel 0 au pixel 7.

c. Registres du séquenceur

Le séquenceur (Sequencer) constitue un bloc hardware avec le contrôleur graphique (Extended Graphics Controller) qui contrôle l'accès à la mémoire vidéo, le rafraîchissement des caractères (clocking) et la cartographie du générateur de caractères. Les registres du séquenceur sont : un registre d'adressage (address) et quatre registres adressés par lui. La position d'indice 0 bien que documentée pour la sélection du registre Reset est inutile dans le PC 1640 et non implémentée. Les autres registres correspondent aux indices 1 à 4, respectivement le Clock Mode Register, le Plane Mask Register, le Character Set Select Register et le Memory Mode Register.

Registre d'adressage du Séquenceur

Le Sequencer Address Register est un registre en écriture seule accessible par l'adresse d'E/S 3C4 qui détermine quel registre du Séquenceur sera rattaché à l'adresse d'E/S 3C5. L'affectation des bits du registre Sequencer Address est la suivante :

Bit	Fonction
7-3	Inutilisés
2-0	Adressage du Séquenceur (sur les bits 2 à 0)

Registre Clock Mode (3C5,1)

Le Clock Mode est un registre en écriture seule qui réside à l'adresse d'E/S 3C5 lorsque le registre Sequencer Address a été configuré pour l'indice 1. Son format est le suivant :

Bit	Fonction
7-4	Inutilisés
3	Fréquence d'horloge d'affichage (Dot clock)
2	Chargement des registres de glissement
1	Inutilisé
0	8/9 dot clocks

Le bit 3 sert à déterminer si la fréquence d'horloge de l'affichage (dot clock) sera identique (si bit 3=0) à celle de l'horloge maîtresse ou si elle sera divisée par deux (bit 3=1). Cela sert aux affichages graphiques en 320 pixels (en largeur).

Quand le bit 2 est à zéro, les registres de décalage vidéo (serial video shift) sont rechargés à chaque impulsion d'horloge. Si le bit 2 est à 1, l'affectation d'une valeur à ces registres ne se fait que toutes les secondes impulsions d'horloge de caractère. Le bit 2 doit être à 0 quand deux des registres de décalage (shift registers) sont chaînés pour les colonnes de caractères sur 16 bits.

Le bit 0 contrôle le nombre de dot clocks (impulsions de l'horloge vidéo) générées par ligne de caractères. Il doit être à 0 en mode monochrome (largeur de caractère sur 9 bits) et à 1 pour les modes caractère couleur (largeur sur 8 bits).

Registre Color Plane Select (3C5,2)

Le Color Plane Select Register est un registre à écriture seule qui réside à l'adresse d'E/S 3C5 quand le registre d'indice contient 2.

Bit	Fonction
7-4	Inutilisés
3	Autorisation d'écriture sur le plan mémoire 3
2	Autorisation d'écriture sur le plan mémoire 2
1	Autorisation d'écriture sur le plan mémoire 1
0	Autorisation d'écriture sur le plan mémoire 0

Mettre les bits 0-3 à 1 autorise l'écriture sur les plans mémoire respectifs. Le CPU peut écrire sur toutes combinaisons de plans de mémoire d'affichage en un cycle d'écriture en validant les bits correspondants. Les bits 0 et 1 ainsi que les bits 2 et 3 doivent avoir la même valeur quand les modes paires/impaires sont sélectionnés (voir registre Memory Mode Select).

Registre Character Set Select (3C5,3)

Le Character Set Select Register est un registre à écriture seule qui réside à 3C5 quand le registre d'indice est à 3. Son format est le suivant :

Bit	Fonction
7-4	Inutilisés
3-2	Sélection du jeu de caractères A
1-0	Sélection du jeu de caractères B

Il y a quatre jeux de caractères alternés par table de caractères, chacun représentant 128 caractères, ce qui permet d'avoir pour un écran donné un nombre de 512 caractères. De plus, le registre IGA Extended Mode (bit 3) permet l'activation d'un jeu de caractères secondaire ce qui donne un total de 1024 caractères disponibles. Quand la sélection de table de caractères alternée est inhibée, la sélection du jeu A revient à la sélection du jeu B (c'est-à-dire que les bits 0 et 1 égalent les bits 2 et 3). Quand elle est

autorisée, le bit 3 de l'attribut (Attribute bit) sélectionne la table de caractères A quand il est à 0 et la table de caractères B quand il est à 1. Le tableau suivant donne les sélections de jeux de caractères en fonction des bits 0-3 :

Jeu de caractères				**Table sélectionnée**	
Sel. A		**Sel. B**		**Bit 3 de l'attribut**	
3	**2**	**1**	**0**	**0**	**1**
0	0	0	0	--	--
0	0	0	1	00	01
0	0	1	0	00	02
0	0	1	1	00	03
0	1	0	0	01	00
0	1	0	1	--	--
0	1	1	0	01	02
0	1	1	1	01	03
1	0	0	0	02	00
1	0	0	1	02	01
1	0	1	0	--	--
1	0	1	1	02	03
1	1	0	0	03	00
1	1	0	1	03	01
1	1	1	0	03	02
1	1	1	1	--	--

Les "—" signifient inhibé et que le bit 3 de l'attribut devient le bit de sélection d'intensité.

Quand le bit 3 du registre Extended Mode est mis à 1, les tables de caractères secondaires sont autorisées depuis le plan 3 et dans ce cas, les numéros de tables de caractères dans le tableau doivent être augmentés de 4. La figure suivante illustre l'organisation complète des jeux de caractères du PC 1640.

	Plan 2	Plan 3
8 K octets	Jeu 0	Jeu 4
8 K octets	Inutilisé	Inutilisé
8 K octets	Jeu 1	Jeu 5
8 K octets	Inutilisé	Inutilisé
8 K octets	Jeu 2	Jeu 6
8 K octets	Inutilisé	Inutilisé
8 K octets	Jeu 3	Jeu 7
8 K octets	Inutilisé	Inutilisé

Registre Memory Mode Select

Le Memory Mode Select Register est un registre à écriture seule qui réside à l'adresse d'E/S 3C5 quand le registre d'indice contient 4. Son format est le suivant :

Bit	Fonction
7-3	Inutilisés
2	Paire/Impaire
1	Mémoire étendue
0	Inutilisé

Le bit 1 indique que 256 K de RAM vidéo sont présents et doit donc toujours être à 1 sur le PC 1640.

Quand le bit 2 est à 1, il sélectionne un mode d'adressage chaîné dans lequel les adresses paires du CPU donnent accès aux jeux de caractères 0 et 2 alors que les adresses impaires du CPU donnent accès aux jeux de caractères 1 et 3. Quand le bit 2 est à zéro, cela sélectionne le mode d'adressage non chaîné et le CPU accède séquentiellement aux données bit map des caractères.

d. Registres du contrôleur graphique

Le contrôleur graphique (Graphics Controller) est un élément hardware de l'Extended graphics Controller qui envoie les données mémoire vers le contrôleur d'attributs et vers le CPU. Les données de la mémoire sont envoyées en série au contrôleur d'attributs lorsqu'on se trouve en mode graphique. Le Graphics Controller et l'Attribute Controller sont deux dipositifs logiques qui forment un ensemble fonctionnel appelé Video Controller. Les registres du contrôleur graphique sont au nombre de dix : un registre d'adressage et neuf registres de données à écriture seule.

Graphics Controller Address (3CE)

Le registre d'adressage du contrôleur graphique est accessible en écriture seule par l'adresse d'E/S 3CE qui sert de pointeur (d'indice) vers les neuf registres de données dans lesquels on écrit via l'adresse d'E/S 3CF. L'attribution des bits est la suivante :

Bit	Fonction
7-4	Inutilisés
3-0	Pointeur vers les registres

Les bits 0-3 sélectionnent le registre courant à l'adresse d'E/S 3DF.

Set/Reset (3CF, 0)

Le registre Set/Reset est un registre à écriture seule auquel on accède par l'adresse d'E/S 3CF quand le registre d'adressage (Graphics Controller Address) contient 0. Son format est le suivant :

Bit	Fonction
7-4	Inutilisés
3	Bit 3 du Set/Reset
2	Bit 2 du Set/Reset
1	Bit 1 du Set/Reset
0	Bit 0 du Set/Reset

La valeur des bits 0 à 3 du registre Set/Reset est écrite, respectivement, sur les plans 0 à 3 à condition que le bit correspondant dans le registre Enable Set/Reset (Activation de Set/Reset, voir le registre suivant) vaille 1 que le registre Graphic Mode 1 soit programmé en mode écriture 0 et enfin que le bit de la donnée à écrire envoyée par le CPU soit à 1.

Par exemple, si l'on désire affecter la couleur cyan clair aux bits 7, 5 et 0 d'une position d'affichage donnée, les bits 3 à 0 du registre Set/Reset reçoivent la valeur binaire 1011 (qui est la valeur (pattern) correspondant au cyan clair). Puis le registre Enable Set/Reset reçoit la valeur 0Fh ce qui permet d'affecter 1010001 (en binaire) à l'adresse de la mémoire vidéo correspondant à la position d'affichage voulue. Les bits contenant 0 seraient inchangés et les pixels correspondant aux bits 7, 5 et 0 prendraient la couleur cyan clair. L'équation permettant d'obtenir les numéros de pixels à partir des numéros de bits est la suivante :

N° de pixel = (7 - N° de bit)

Registre Enable Set/Reset (3CF,1)

Il s'agit d'un registre accessible en écriture seule par l'adresse d'E/S 3CF quand le registre Graphics Controller Address contient 1. Son format est le suivant :

Bit	Fonction
7-4	Inutilisés
3	Bit 3 du Enable Set/Reset
2	Bit 2 du Enable Set/Reset
1	Bit 1 du Enable Set/Reset
0	Bit 0 du Enable Set/Reset

Mettre à 1 l'un des bits 0 à 3 valide l'écriture (vers l'affichage) du bit correspondant du registre Set/Reset (Enable veut dire autoriser, activer). Si l'on se trouve en mode d'écriture 0 et si Set/Reset est inhibé sur le plan de couleur, le contenu original du plan de couleur est préservé.

Registre Color Plane Compare (3CF,2)

Le registre Color Plane Compare est accessible en écriture seule à l'adresse d'E/S 3CF quand le registre d'adressage du contrôleur graphique (Graphics Controller Address Register) contient 2. Le registre Color Plane Compare est activé quand le bit 3 du registre Graphics Mode 1 (voir plus loin) est mis à 1 ce qui active le mode comparaison. Quand ce mode comparaison (de couleur) est actif, une lecture de la RAM vidéo par le CPU lui retournera le résultat d'une comparaison entre le registre Color Plane Compare et les plans de couleur plutôt que le contenu réel de la mémoire.

Le registre Color Plane Compare est ainsi défini :

Bit	Fonction
7-4	Inutilisés
3	Color Plane Compare 3
2	Color Plane Compare 2
1	Color Plane Compare 1
0	Color Plane Compare 0

Le motif (pattern) des bits 0 à 3 est comparé à l'adresse mémoire lue par le CPU. Les bits retournés représentent ceux des bits qui correspondent réellement au motif. Le registre Color No Care (décrit plus loin) sélectionne les plans participant à la comparaison.

Dans l'exemple d'écriture de pixels cyan clair que nous donnions à propos du registre Set/Select, affecter la valeur binaire 1011 au registre Color Plane Compare et lire la même adresse de la RAM vidéo retournerait une valeur binaire écrite de 1010001. On interprèterait cette réponse comme voulant dire que les pixels correspondant aux bits 7, 5 et 0 sont cyan clair et que les autres pixels sont d'une ou de plusieurs autres couleurs. Connaître les autres couleurs présentes dans cet octet demanderait de procéder à cette comparaison avec les 15 autres couleurs possibles ou, plus sobrement, de lire chaque plan de couleur et d'en analyser la couleur effectivement définie par les quatre bits représentant chaque pixel.

Registre Data Rotate (3CF,3)

Le registre Data Rotate est un registre à écriture seule qui réside à l'adresse d'E/S 3CF quand le registre Graphics Controller Address contient 3. Ce registre contrôle un certain nombre d'opérations logiques exécutées quand le CPU écrit dans la RAM vidéo. Il est ainsi défini :

Bit	Fonction
7-5	Inutilisés
4-3	Sélecteur de fonction
2-0	Compteur de rotation

Le champ de sélection de la fonction (bit 3 et 4) permet de choisir l'une des quatre fonctions logiques opérant sur le contenu existant d'un plan de mémoire et la nouvelle donnée à écrire. Ces opérations sont les suivantes :

Bit 4	Bit 3	Fonction sélectionnée
0	0	Remplacement des données existantes
0	1	AND logique
1	0	OR logique
1	1	XOR logique

Quand les bits 3 et 4 sont tous deux à 0, la donnée provenant du CPU remplace purement et simplement la valeur pré-existante sans qu'il ne leur soit appliqué de traitement logique.

Lorsqu'une opération logique est active, les plans sur lesquels l'écriture est autorisée (voir le registre Color Plane Enable) font l'objet d'une lecture, puis d'une opération booléenne avec la donnée provenant du CPU. La valeur résultante est affectée à la mémoire des plans d'affichage respectifs.

Les bits 0 à 3 contrôlent une rotation à droite (Rotate Right) effectuée sur les données à écrire envoyées par le CPU. Quand le mode d'écriture 0 est programmé dans le registre Graphics Mode 1 (décrit plus loin) le contenu des bit 0 à 3 représente le nombre de rotations effectuées sur la valeur venant du CPU. Durant une rotation, le bit de poids faible "écrit" dans le bit de poids fort en décalant ce dernier.

Le résultat est affecté au(x) plan(s) sélectionné(s).

Registre Color Plane Read (3CF,4)

Le registre Color Plane Read est un registre à écriture seule qui réside à l'adresse d'E/S 3CF quand le registre Graphics Controller Address contient 4. Son format est le suivant :

Bit	Fonction
7-2	Inutilisés
1-0	Sélection mode lecture

Les lectures du CPU auront lieu sur l'un des quatre plans sélectionnés par la valeur des bits 0 et 1 (nom anglais Read Select). Si le contenu original des registres de palette est maintenu, la sélection a trait aux signaux couleur IRGB de la façon suivante :

Bits 0	1	Plan de couleur sélectionné
0	0	Bleu (Plan 0)
0	1	Vert (Plan 1)
1	0	Rouge (Plan 2)
1	1	Intensité (Plan 3)

Le bit 2 est aussi connecté au décodage du sélecteur de lecture et doit être maintenu à 0 sinon aucun plan ne sera sélectionné pour la lecture.

Registre Graphics Mode 1 (3CF,5)

Le registre Graphics Mode (mode graphique) est un registre en écriture seule qui réside à l'adresse d'E/S 3CF quand le registre

d'adressage du contrôleur graphique a pour valeur 5. Son format est le suivant :

Bit	Fonction
7-6	Inutilisés
5	Format du registre de décalage (Shift Register)
4	Inutilisé
3	Mode lecture
2	Inutilisé
1-0	Mode d'écriture

Le bit 5 contrôle l'interaction des quatre registres de décalage chaînés en série et du contrôleur graphique. En temps normal, quand le bit 5 est à 0, les données provenant des plans 0-3 "sortent" des registres respectifs de décalage 0-3 comme si l'on effectuait un glissement vers l'extérieur avec le bit de poids fort sortant en tête. Quand le bit 5 est mis à 1, les bits portant un numéro pair font l'objet d'un glissement vers l'extérieur à partir des registres de décalage portant un numéro pair. Les bits portant un numéro impair proviennent de la même façon des registres de décalage portant un numéro impair. Cela signifie que le registre de décalage (shift register) 0 décale les bits 6, 4, 2 et 0 du plan 0 suivis par les bits 6, 4, 2 et 0 du plan 1. Le registre de décalage 1 décale les bits 7, 5, 3 et 1 du plan 0 suivis des bits 7, 5, 3 et 1 du plan 1. Les registres de décalage 2 et 3 agissent avec le même modèle pair/impair sur les plans de couleur 2 et 3.

Lorsque le bit 3 est à 0, il autorise le registre Color Plane Read à contrôler la sélection des plans de couleurs lus par le CPU. Lorsque le bit 3 est à 1, c'est le registre Color Plane Compare qui intervient et qui donc filtre les lectures du CPU comme nous l'avons expliqué dans la section consacrée à ce registre Color Plane Compare.

Les bits 0 et 1 forment un champ de deux bits spécifiant le mode d'écriture (Write Mode). Cela détermine la manière dont l'IGA manipule les données graphiques écrites sur les plans mémoire.

Le tableau suivant recense les options possibles :

Bits 1	0	Description du mode d'écriture sélectionné
0	0	Les données du CPU écrites sur les plans 0-3 sont contrôlées par le registre Color Plane Write, le registre Data Rotate, le registre Set/Reset comme indiqué dans la description de ces registres.
0	1	Lorsqu'un cycle d'écriture du CPU est lancé, ce sont les données lues lors de la précédente opération de lecture du CPU qui sont écrites sur les plans 0 à 3.
1	0	Les plans 0-3 reçoivent des 1 ou des 0 en fonction des bits 0-3 de la donnée du CPU. Par exemple, si le bit 3 de la donnée est à 1, on écrit FFh sur le plan 3.

L'état 1, 1 est illégal et ne doit pas être employé.

Registre Graphics Mode 2 (3CF,6)

Ce second registre de mode graphique (Graphics Mode Register 2) est un registre à écriture seule qui réside à l'adresse d'E/S 3CF quand le registre Graphics Controller Address est à 6. Son format est le suivant :

Bit	Fonction
7-4	Inutilisés
3-2	Mode d'utilisation de la mémoire (mapping)
1	Activation du chaînage Impair/Pair
0	Inutilisé

Les bits 2 et 3 forment un champ qui détermine les paramètres concernant l'origine et la taille du tampon de régénération.

Leurs valeurs correspondent aux configurations suivantes :

Bit 3	Bit 2	Origine	Taille
0	0	A0000h	128 K octets
0	1	A0000h	64 K octets
1	0	B0000h	32 K octets
1	1	B8000h	32 K octets

Mettre le bit 1 à 1 fait que le bit d'adresse A0 contrôle la sélection des plans mémoire impairs/pairs plutôt que les adresses contigües impaires/paires. La fonction habituelle réalisée par le bit d'adresse A0 est décalée vers le bit d'adresse d'ordre supérieur.

Registre Color No Care (3CF, 7)

Ce registre est accessible en écriture seule par l'adresse d'E/S 3CF lorsque le registre d'adressage du contrôleur graphique contient 7. Son format est le suivant :

Bit	Fonction
7-4	Inutilisés
3	Négliger le plan de couleur 3
2	Négliger le plan de couleur 2
1	Négliger le plan de couleur 1
0	Négliger le plan de couleur 0

Ce registre est lié au registre Color Compare et peut être considéré comme un masque des comparaisons de couleurs en cela que lorsqu'un des bits 0-3 est à 1, le plan de couleurs dont le numéro correspond n'est pas pris en compte dans l'opération de comparaison.

Registre Write Mask (3CF, 8)

Le registre Write Mask est à écriture seule et l'on y accède par l'adresse d'E/S 3CF lorsque le Graphics Controller Address contient 8.

Bit	Fonction
7-0	Masque d'écriture (bits 7-0)

Lorsqu'un bit du Write Mask (masque d'écriture) est à 0, il interdit l'écriture sur le plan d'affichage du bit correspondant de la donnée. Le masque d'écriture est programmé une fois pour toutes durant l'initialisation du mode d'affichage de sorte que tous les bits sont normalement écrits sur la RAM vidéo.

Ce registre Write Mask affecte toutes les valeurs écrites par le CPU y compris celles faisant l'objet de rotations ou d'opérations logiques. Le hardware effectue une lecture-avant-écriture afin de préserver les bits protégés.

e. Registres du contrôleur CRT en mode EGA

Le contrôleur CRT est une composante hardware de l'IGA qui contrôle la synchronisation horizontale et verticale, le curseur, le soulignage et la fréquence de clignotement. Il génère aussi l'adressage du tampon de régénération de l'affichage et les contrôles de rafraîchissement dynamique de la RAM.

Le contrôleur CRT comporte 27 registres de données auxquels on accède par un registre d'indice. Ce dernier doit être chargé en premier lieu pour pouvoir accéder à un registre CRTC particulier. Les registres de données du CRTC résident soit à l'adresse d'E/S 3B5, soit à l'adresse d'E/S 3D5 selon que, respectivement, l'IGA opère en mode monochrome ou couleur.

Ce schéma d'adressage est ainsi conçu pour que l'installation d'une seconde carte graphique dans un slot d'extension ne génère pas de conflits d'adresses pour les entrées/sorties avec le contrôleur vidéo principal (CRTC). Cela signifie aussi que le système peut accepter au maximum une carte monochrome et une carte couleur simultanément. Si l'IGA pilote un moniteur monochrome, on peut placer une carte CGA dans un slot. Réciproquement, si l'IGA contrôle un moniteur couleur, on ne pourra installer qu'une carte monochrome dans les slots. En aucun cas, il n'est question d'ajouter une carte EGA dans un des slots lorsque l'IGA est actif puisque l'autre EGA va écraser le BIOS en ROM de l'IGA ainsi que les registres dont l'adresse d'E/S se trouve dans la zone 3CX. On peut aussi configurer l'IGA de manière à ce qu'il recouvre la RAM vidéo de l'autre adaptateur.

Registre CRT Controller Address

Le registre CRT Controller Address est un registre à écriture seule auquel on accède soit par l'adresse d'E/S 3B4, soit par l'adresse d'E/S 3D4. La valeur qu'on y écrit doit représenter un indice des registres de données du CRTC à l'adresse d'E/S 3X5 (X = B pour Monochrome et D pour couleur).

Bit	Fonction
7-4	Inutilisés
4-0	Adressage des registres CRTC (00h-18h)

Les valeurs supérieures à 18h sont invalides et ne doivent pas être utilisées.

Registre Horizontal Total (3B5/3D5,0)

Le registre Horizontal Total est un registre en écriture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 00h. Son format est le suivant :

Bit	Fonction
7-0	Valeur du total horizontal (-2)

Le registre Horizontal Total spécifie en nombre de caractères (moins 2) l'intervalle de balayage horizontal y compris le retour. Cette valeur est la base des timings horizontaux et verticaux.

Registre Horizontal Display End (3B5/3D5,1)

Le registre Horizontal Display End (Fin d'affichage horizontal) est un registre en écriture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 01h. Son format est le suivant :

Bit	Fonction
7-0	Valeur de la fin d'affichage horizontal (-1)

Cette valeur est le nombre de caractères à afficher par ligne horizontale. Le nombre réel de caractères affichés par ligne horizontale est inférieur de 1 au contenu de ce registre.

Registre Start Horizontal Blanking (3B5/3D5,2)

Le registre Start Horizontal Blanking (début d'effacement horizontal) est un registre en écriture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 02h. Il est formé comme suit :

Bit	Fonction
7-0	Valeur du départ horizontal d'effacement

Le signal d'effacement horizontal devient actif quand le comptage des caractères horizontaux est égal à la valeur de ce registre.

Registre End Horizontal Blanking (3B5/3D5,3)

Le registre End Horizontal Blanking (fin d'effacement horizontal) est un registre en écriture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 03h. Son format est le suivant :

Bit	Fonction
7	Inutilisé
6-5	Valeur du décalage de l'activation d'affichage
4-0	Valeur de la fin d'effacement horizontal

Le signal d'effacement horizontal (horizontal blanking) devient actif quand les 5 bits de poids faible du compteur de caractères horizontaux sont égaux à la valeur représentée par les bits 0-4 de ce registre.

A cause de l'accès séquentiel du CRTC à la mémoire vidéo, les données vidéo sont décalées par rapport au timing horizontal. La valeur de décalage d'activation de l'affichage (Display Enable Skew) y remédie en imposant un délai en temps de caractères égal à la valeur rangée dans les bits 5 et 6.

L'équation suivante définit la valeur des bits 0-4 :

Fin d'effacement horizontal = (Valeur de départ d'effacement + Largeur de l'effacement) Modulo 32

Registre Start Horizontal Retrace (3B5/3D5,4)

Le registre Start Horizontal Retrace est un registre en écriture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 04h. Son format est le suivant :

Bit	Fonction
7-0	Valeur du départ de retraçage

Le signal de retraçage horizontal est activé quand le compteur de caractères horizontaux atteint la valeur de ce registre.

Registre End Horizontal Retrace (3B5/3D5,5)

Le registre End Horizontal Retrace est un registre en écriture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 05h. L'assignation des bits est la suivante :

Bit	Fonction
7	Adresse de départ Paire/Impaire
6-5	Décalage de retraçage horizontal
4-0	Valeur de la fin de retraçage horizontal

Le signal de retraçage horizontal devient passif quand les 5 bits de poids faible du compteur de caractères horizontaux prennent la valeur des bits 0-4 de ce registre.

Si le bit 7 est à 1, cela spécifie que l'adresse mémoire CRT sera une adresse impaire après un retraçage horizontal. Mettre le bit 7 à 0 spécifie une adresse paire. En général, ce bit doit être à 0 mais il peut être utile pour certaines applications où un panning (glissement) horizontal des pixels est nécessaire.

La valeur des bits 5 et 6 détermine le temps de décalage du retraçage horizontal (0-3) exprimé en temps de caractères. Le signal de retraçage horizontal sera retardé de cette valeur. Un certain nombre de signaux de tempo internes sont générés à partir du front descendant du retraçage horizontal. Pour garantir un bon verrouillage (latching), le retraçage commence avant la fin d'activation d'affichage et est ensuite retardé (décalé) de plusieurs caractère-temps afin d'obtenir un centrage de l'écran correct.

L'équation suivante permet de déterminer la valeur des bits 0-4 :

Fin de retraçage horizontal = (Valeur du départ de retraçage + Largeur du retraçage) Modulo 32

Registre Vertical Total (3B5/3D5,6)

Le registre Vertical Total est un registre en écriture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 06h. Son format est le suivant :

Bit	Fonction
0-7	Valeur du total vertical (-2)

Ce registre spécifie les 8 bits de poids faible d'une valeur de 9 bits. Le neuvième bit est situé dans le registre CRTC Overflow. La valeur de ce registre représente le nombre total de lignes de balayage par trame moins 2.

Registre CRTC Overflow (3B5/3D5,7)

Le registre CRTC Overflow est un registre en écriture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 07h. Son format est le suivant :

Bit	Fonction
7-5	Inutilisés
4	Bit 8 de Line Compare
3	Bit 8 de Start Vertical Blanking
2	Bit 8 de Start Vertical Retrace
1	Bit 8 de Vertical Display Enable
0	Bit 8 de Vertical Total

Ce registre renferme le neuvième bit de plusieurs autres registres de contrôle qui sera mis à 1 si l'on spécifie une ligne de balayage (scan line) supérieure à 255.

Registre Preset Row Scan (3B5/3D5,8)

Le registre Preset Row Scan (pré-affectation de valeur de ligne de balayage) est un registre en écriture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 08h. Son format est le suivant :

Bit	Fonction
7-5	Inutilisés
4-0	Valeur de ligne de balayage.

Les bits 0-4 déterminent la valeur du compteur de départ de ligne de balayage après un retraçage vertical.

Registre Maximum Scan Line (3B5/3D5,9)

Le registre Maximum Scan Line (ligne de balayage maximum) est un registre en écriture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 09h. L'assignation des bits est la suivante :

Bit	Fonction
7-5	Inutilisés
4-0	Ligne de balayage maximum (-1)

La valeur programmée dans les bits 0-4 définit le nombre de lignes de balayage par caractère moins une.

Chaque retraçage horizontal incrémente le compteur de lignes de balayage horizontal. Le compteur de lignes de balayage horizontal est remis à 0 quand il arrive à la valeur du registre Maximum Scan Line.

Registre Cursor Start (3B5/3D5,0A)

Le registre Cursor Start est un registre en écriture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 0Ah. Son format est le suivant :

Bit	Fonction
7-6	Inutilisés
5	Curseur invisible
4-0	Ligne de départ du curseur (-1)

Mettre le bit 5 à 1 inhibe l'affichage du curseur. Mettre le bit 5 à 0 active l'affichage du curseur.

Les bits 0-4 (qu'on appelle départ du curseur par analogie avec le contrôleur vidéo MC845 et donc à ne pas confondre avec le registre Cursor Start) contiennent le numéro de ligne de balayage à l'intérieur d'une ligne de caractère où le curseur va commencer. Donner à ces 5 bits une valeur supérieure à la fin de curseur inhibe l'affichage du curseur.

Registre Cursor End (3B5/3D5,0B)

Le registre Cursor End est un registre en écriture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 0Bh. Son format est le suivant :

Bit	Fonction
7	Inutilisé
6-5	Contrôle du délai (décalage) du curseur
4-0	Ligne d'arrivée du curseur (+1)

La valeur contenue dans les bits 0-4 représente la dernière ligne de balayage d'une ligne de caractère où se fera l'affichage du curseur plus une.

Les bits 5 et 6 spécifient le nombre de caractères-temps constituant le délai d'affichage du curseur. Pour chaque caractère-temps, le curseur est décalé à droite d'une position.

La programmation des registres Cursor Start et Cursor End ne donne pas toujours le résultat attendu en particulier dans les cas où l'on donne à la fin de curseur une valeur supérieure au nombre de lignes de balayage par caractère.

Registre Start Address High (3B5/3D5,0C)

Le registre Start Address High (adresse de départ haute) est un registre en lecture/écriture qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 0Ch. L'assignation de ses bits est la suivante :

Bit	Fonction
7-0	Adresse de départ de l'affichage (Bits de poids fort)

Les registres Start Address High et Start Address Low forment une valeur 16 bits qui est ajoutée à l'origine de l'affichage programmée par le registre Graphics Mode 2 pour former l'adresse de début du tampon d'affichage courant. Dans les modes textes, le BIOS IGA en ROM programme le CRTC de manière à ce que l'adresse de départ de l'affichage définisse des paires d'octets caractère/attribut accessibles par une simple multiplication par 2 (décalage) du numéro de paire pour obtenir l'offset à partir de l'adresse d'origine de l'affichage ; ce qui ramène à une mémoire organisée en mots. Dans les modes graphiques, la valeur d'offset correspond au rang d'octet.

Registre Start Address Low (3B5/3D5,0D)

Le registre Start Address Low est un registre en lecture/écriture qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 0Dh. Les bits sont assignés comme suit :

Bit	Fonction
7-0	Adresse de départ de l'affichage (Bits de poids faible)

Ce registre spécifie les 8 bits de poids faible (les autres 8 bits, de poids fort, sont ceux de Start Address High) de l'adresse de départ de l'affichage.

Registre Cursor Location High (3B5/3D5,0E)

Le registre Cursor Location High (poids fort de l'emplacement du curseur) est un registre en lecture/écriture qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 0Eh. Son format est le suivant :

Bit	Fonction
7-0	Emplacement du curseur (bits de poids fort)

La position du curseur est une valeur sur 16 bits similaire à celle du registre Start Address. Le registre Cursor Location High contient donc les 8 bits de poids fort.

Registre Cursor Location Low (3B5/3D5,0F)

Le registre Cursor Location Low (poids faible de l'emplacement du curseur) est un registre en lecture/écriture qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 0Fh. Son format est le suivant :

Bit	Fonction
7-0	Position du curseur (Bits de poids faible)

Le registre Cursor Location Low contient les 8 bits de poids faible de l'emplacement du curseur.

Registre Start Vertical Retrace (3B5/3D5,10 Ecriture)

Le registre Start Vertical Retrace (départ du retraçage) est un registre en écriture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 10h. Son format est le suivant :

Bit	Fonction
7-0	Position du départ de retraçage (8 bits de poids faible)

Le registre Start Vertical Retrace est un registre 9 bits et sa valeur spécifie la position du front du signal de retraçage vertical exprimé en lignes de balayage. Ce registre comprend les 8 bits faibles de la valeur réelle, le bit de poids fort (le neuvième) se trouve dans le registre CTRC Overflow.

Registre End Vertical Retrace (3B5/3D5,11 Ecriture)

Le registre End Vertical Retrace (Fin de retraçage vertical) est un registre en écriture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 11h. Il possède le format suivant :

Bit	Fonction
7-6	Inutilisés
5	Activation de Vertical Interrupt
4	Annulation de Vertical Interrupt
3-0	Valeur de la fin de retraçage vertical

Le signal de retraçage vertical devient inactif quand les quatre bits de poids faible du compteur de balayage horizontal prennent une valeur égale à celle des bits 0-3 de ce registre.

Quand le bit 5 est à 1, la broche IRQ prend un état haut-Z triple et quand il est à 0, l'IGA génère IRQ2 dès que le retraçage vertical devient actif.

Le bit est mis à 0 pour annuler le signal d'interruption du retraçage vertical. La routine de traitement de l'interruption du CRTC doit mettre ce bit à 0 puis le remettre à 1 pour effacer l'interruption.

L'équation suivante permet de déterminer la valeur des bits 0-3 :

Fin de retraçage vertical = (Valeur du départ de retraçage vertical + largeur du retraçage) Modulo 16

Registre Light Pen High (3B5/3D5,10 lecture)

Le registre Light Pen High (crayon optique) est un registre en lecture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 10h. Son format est le suivant :

Bit	Fonction
7-0	Adresse du crayon optique (bits de poids fort)

L'adresse du crayon optique est une valeur 16 bits égale à la valeur du registre d'adresse du CRTC au moment où le crayon a été activé. Ce registre en retourne les huit bits de poids fort.

Registre Light Pen Low (3B5/3D5,11 lecture)

Le registre Light Pen Low (crayon optique) est un registre en lecture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 11h. Son format est le suivant :

Bit	Fonction
7-0	Adresse du crayon optique (bits de poids faible)

Une opération de lecture d'E/S de ce registre retourne les 8 bits de poids faible de la valeur d'adresse de l'activation du crayon optique.

Registre Vertical Display End (3B5/3D5,12)

Le registre Vertical Display End (fin d'affichage vertical) est un registre en écriture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 12h. Son format est le suivant :

Bit	Fonction
7-0	Valeur de fin d'affichage vertical (-1) (8 bits de poids faible)

La valeur de fin d'activation de l'affichage représente le nombre de lignes de balayage affichées moins une. Ce registre contient les 8 bits de poids faible, le neuvième bit se trouvant dans le registre CRTC Overflow.

Registre CRTC Offset (3B5/3D5,13)

Le registre CRTC Offset est un registre en écriture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 13h. Son format est le suivant :

Bit	Fonction
7-0	Valeur de l'offset CRTC

La valeur de l'offset CRTC représente la largeur logique d'une ligne de l'écran. La valeur de ce registre détermine l'adresse de départ de la prochaine ligne de caractères et est le nombre de paires caractère/attribut par ligne affichée.

Registre Underline Location (3B5/3D5,14)

Le registre Underline Location (position du souligné) est un registre en écriture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 14h. Son format est le suivant :

Bit	Fonction
7-5	Inutilisés
4-0	Valeur de la position du soulignement (-1)

La valeur des bits 0-4 représente le numéro de ligne de balayage horizontale où le soulignement se produira.

Cette valeur est de 1 plus petite que la ligne de balayage effectivement désirée.

Registre Start Vertical Blanking (3B5/3D5,15)

Le registre Start Vertical Blanking (départ de l'effacement vertical) est un registre en écriture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 15h.

Bit	Fonction
7-0	Valeur du départ de l'effacement vertical (8 bits de poids faible)

Ce registre contient les 8 bits de poids faible du numéro de ligne de balayage horizontal où se produira l'activation du signal d'effacement vertical. Le neuvième bit de cette valeur est situé dans le registre CRTC Overflow.

Registre End Vertical Blanking (3B5/3D5,16)

Le registre End Vertical Blanking (fin d'effacement vertical) est un registre en écriture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 16h. Son format est le suivant :

Bit	Fonction
7-5	Inutilisés
4-0	Valeur de fin d'effacement vertical

Le signal d'effacement vertical est activé quand les 5 bits de poids faible du compteur de lignes de balayage horizontal prennent la même valeur que les bits 0-4 de ce registre.

L'équation suivante indique la valeur des bits 0-4 :

Fin d'effacement vertical = (Départ d'effacement vertical + Largeur de l'effacement) Modulo 32

Registre CRTC Mode (3B5/3D5,17)

Le registre CRTC Mode est un registre en écriture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 17h. Son format est le suivant :

Bit	Fonction
7	Activation du retraçage H/V
6	Mode Mot/Octet
5	Recouvrement d'adresses 64 K
4	Inutilisé
3	Mode du compteur d'adresses
2	Sélection de l'horloge de retraçage horizontal
1	Sélection du compteur de lignes de balayage
0	Sélection du mode compatible 200 lignes

Le bit 7 détermine si le retraçage horizontal et vertical est activé ou inhibé. Mettre le bit 7 à 0 inhibe le retraçage horizontal et le retraçage vertical. Mettre le bit 7 à 1 autorise l'émission des signaux de retraçage H et V.

Le bit 6 mis à 0 sélectionne le mode mot (16 bits) et mis à 1 sélectionne le mode octet (8 bits). Ce bit doit être à 0 dans tous les modes sauf EGA (bit 6 du registre IGA Extended Mode à 1). En mode mot, la mémoire est adressée par paire d'octets et le bit de poids fort du compteur d'adresse est attaché au bit d'adresse 0.

Le bit 5 sélectionne le mode recouvrement d'adresses par blocs de 64 K. Etant donné que la carte mère dispose de 256 K de mémoire vidéo, ce bit doit être mis à 1.

Lorsque le bit 3 est à 1, le compteur d'adresses est cadencé au rythme d'un temps-caractère sur deux. Quand le bit 3 est à 0, le compteur d'adresse est directement cadencé par l'entrée de l'horloge caractère.

Le bit 2 permet de déterminer si le compteur de timing vertical est cadencé par le retraçage horizontal ou par le retraçage horizontal divisé par deux. Ce dernier mode, actif quand le bit 2 est à 1, permet de doubler la résolution verticale.

Le bit 1 permet de sélectionner la configuration de la sortie du compteur de lignes de balayage. Mis à zéro, le bit 1 du compteur est placé dans le bit 14 de l'adresse de sortie. Mis à 1, le bit 14 du compteur est placé dans le bit 1 de l'adresse de sortie.

Le bit 0 permet d'offrir la compatibilité CGA en faisant en sorte que les lignes d'affichages alternées soient déplacées de 8 K. Quand le bit 0 est à zéro, le mode compatible est sélectionné et le bit 0 est connecté au bit 13 de l'adresse mémoire durant le temps où l'affichage est actif. Quand le bit 0 est à un, le bit 13 de l'adresse mémoire est connecté au bit 13 du compteur d'adresses.

Registre Line Compare (3B5/3D5,18)

Le registre Line Compare est un registre en écriture seule qui réside à l'adresse d'E/S 3X5 quand le registre CRT Controller Address contient 18h. Son format est le suivant :

Bit	Fonction
7-0	Valeur de comparaison (8 bits de poids faible)

Ce registre contient les 8 bits de poids faible (le neuvième bit est dans le registre CRTC Overflow) de la valeur servant de modèle de comparaison qui permet d'implémenter une fonction de partage d'écran où certaines zones peuvent être protégées du défilement

(scrolling). Quand le compteur de lignes de balayage vertical arrive à la valeur de ce registre, le générateur d'adresses est mis à 0. Chaque adresse de ligne qui en découle est déterminée par l'addition sur 16 bits du latch (verrou) de départ de ligne et du contenu du registre Offset.

Registre Status Port (3BA/3DA)

Selon la valeur du bit 0 (CRTC Mono/Couleur) du registre IGA External Control, on accède au registre Status Port (port d'état) par l'adresse d'E/S 3BA ou 3DA. Son format est le suivant :

Bit	Fonction
7	Inutilisé (Couleur)/-VSYNC (Mono)
6	Mode EGA
5-4	Auto-test Couleur (MUX)
3	-VSYNC (Couleur)/Vidéo Mono
2	Bouton du crayon optique
1	Détection du crayon optique
0	Activation de l'affichage

Le bit 7 est toujours à 1 en mode émulation EGA (Bit 6 du registre IGA Extended Mode = 0) ou quand on se trouve en configuration mémoire de la zone de l'adaptateur couleur (3DA).

En mode émulation MDA (à 3BA), le bit 7 = 0 quand VSYNC est active est le bit 7 = 1 quand VSYNC est inactive.

Le bit 6 est à 1 en mode émulation EGA et à 0 autrement.

Les bits 4 et 5 servent aux programmes d'auto-test pour tester en sortie les signaux couleur et sont utilisés en conjonction avec l'état vidéo MUX (Video Status) du registre Color Plane Enable. Reportez-vous à la description de ce registre pour plus de détails. Nous répétons ici le tableau Video MUX :

Video Status MUX (3C0) Bits 5 4	Reg. Status Port (3XA) Bits 5 4
0 0	R B
0 1	r g
1 0	b G

La lettre G ou g représente Vert ou vert.

Les lettre majuscules représentent les couleurs primaires et les lettres minuscules les couleurs secondaires. Le cas 1, 1 n'est pas utilisé.

En mode EGA ou couleur (3DA), le bit 3 vaut 0 quand VSYNC est active et 1 quand VSYNC est inactive. En mode MDA (3BA), le bit 3 vaut 0 quand le mode Vidéo Mono est actif et 1 quand le mode Vidéo Mono est inactif.

Le bit 2=0 quand le commutateur du crayon optique est fermé et 1 quand il est ouvert.

Le bit 1=0 quand la détection du crayon optique est active ; le bit 1=1 quand cette détection est inactive.

Le bit 0=0 quand le signal vidéo est activé et bit 0=1 quand le signal de retraçage vertical ou horizontal est actif. C'est un flag en temps réel de l'état de la ligne de balayage dans l'affichage d'une ligne d'écran.

1.11.9 L'initialisation du mode EGA de l'IGA BIOS

La ROM du BIOS IGA comprend des tables d'initialisation pour les registres du mode compatible EGA. Les modes du BIOS IGA ont été présentés dans la section 1.11.7 et consistent en un ensemble standard du hardware pour les modes accessibles à la ROM BIOS. Bien évidemment, d'autres configurations logicielles du hardware pourraient être imaginées par un ingénieur système.

Cependant, les modes de fonctionnement de la ROM BIOS offrent une garantie de portabilité vers d'autres matériels compatibles. Les valeurs affectées par le BIOS sont les suivantes :

Mode		0	1	2	3	4	5	6	7	13	14	15	16
EGC cont (3C2)		23	23	23	23	23	23	23	A6	23	23	A2	A7
Registres	(0)	00	00	00	00	00	00	00	00	00	00	00	00
controle	(1)	01	01	01	01	13	13	17	08	01	01	08	01
attributs	(2)	02	02	02	02	15	15	17	08	02	02	00	02
(3C0)	(3)	03	03	03	03	17	17	17	08	03	03	00	03
	(4)	04	04	04	04	02	02	17	08	04	04	18	04
	(5)	05	05	05	05	04	04	17	08	05	05	18	05
	(6)	06	06	06	06	06	06	17	08	06	06	00	14
	(7)	07	07	07	07	07	07	17	08	07	07	00	07
	(8)	10	10	10	10	10	10	17	10	10	10	00	38
	(9)	11	11	11	11	11	11	17	18	11	11	08	39
	(0A)	12	12	12	12	12	12	17	18	12	12	00	3A
	(0B)	13	13	13	13	13	13	17	18	13	13	00	3B
	(0C)	14	14	14	14	14	14	17	18	14	14	00	3C
	(0D)	15	15	15	15	15	15	17	18	15	15	18	3D
	(0E)	16	16	16	16	16	16	17	18	16	16	00	3E
	(0F)	17	17	17	17	17	17	17	18	17	17	00	3F
	(10)	08	08	08	08	01	01	01	0E	01	01	0B	01
	(11)	00	00	00	00	00	00	00	00	00	00	00	00
	(12)	0F	0F	0F	0F	03	03	01	0F	0F	0F	05	0F
	(13)	00	00	00	00	00	00	00	08	00	00	00	00
Sequenceur	(1)	0B	0B	01	01	0B	0B	01	00	0B	01	01	01
	(2)	03	03	03	03	03	03	01	03	0F	0F	0F	0F
(3C5)	(3)	00	00	00	00	00	00	00	00	00	00	00	00
	(4)	03	03	03	03	02	02	06	03	06	06	06	06
Registres	(0)	00	00	00	00	00	00	00	00	00	00	00	00
controle	(1)	00	00	00	00	00	00	00	00	00	00	00	00
graphique	(2)	00	00	00	00	00	00	00	00	00	00	00	00
(3CF)	(3)	00	00	00	00	00	00	00	00	00	00	00	00
	(4)	00	00	00	00	00	00	00	00	00	00	00	00
	(5)	10	10	10	10	30	30	00	10	00	00	00	00
	(6)	0E	0E	0E	0E	0F	0F	0D	0A	05	05	05	05
	(7)	00	00	00	00	00	00	00	00	0F	0F	0F	0F
	(8)	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

BIOS Mode		0	1	2	3	4	5	6	7	13	14	15	16
CRTC	(0)	37	37	70	70	37	37	70	60	37	70	60	5B
	(1)	27	27	4F	4F	27	27	4F	4F	27	4F	4F	4F
(3B5/3D5)	(2)	2D	2D	5C	5C	2D	2D	59	56	2D	59	56	53
	(3)	37	37	2F	2F	37	37	2D	3A	37	2D	3A	37
	(4)	31	31	5F	5F	30	30	5E	51	30	5E	50	52
	(5)	15	15	07	07	14	14	06	60	14	06	60	00
	(6)	04	04	04	04	04	04	04	70	04	04	70	6C
	(7)	11	11	11	11	11	11	11	1F	11	11	1F	1F
	(8)	00	00	00	00	00	00	00	00	00	00	00	00
	(9)	07	07	07	07	01	01	01	0D	00	00	00	00
	(0A)	06	06	06	06	00	00	00	0B	00	00	00	00
	(0B)	07	07	07	07	00	00	00	0C	00	00	00	00
	(0C)	00	00	00	00	00	00	00	00	00	00	00	00
	(0D)	00	00	00	00	00	00	00	00	00	00	00	00
	(0E)	00	00	00	00	00	00	00	00	00	00	00	00
	(0F)	00	00	00	00	00	00	00	00	00	00	00	00
	(10)	E1	E1	E1	E1	E1	E1	E0	5E	E1	E0	5E	5E
	(11)	24	24	24	24	24	24	23	2E	24	23	2E	2B
	(12)	C7	C7	C7	C7	C7	C7	C7	5D	C7	C7	5D	5D
	(13)	14	14	28	28	14	14	28	28	14	28	28	28
	(14)	08	08	08	08	00	00	00	0D	00	00	0D	0F
	(15)	E0	E0	E0	E0	E0	E0	DF	5E	E0	DF	5E	5F
	(16)	F0	F0	F0	F0	F0	F0	EF	6E	F0	EF	6E	0A
	(17)	A3	A3	A3	A3	A2	A2	C2	A3	E3	E3	E3	E3
	(18)	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

1.11.10 Les registres compatibles CGA

Quand le PC 1640 est en mode compatible Color Graphics Adapter (CGA), et que l'on dispose d'un écran PC-CD (ou PC-ECD), on obtient donc l'ancien mode texte et graphique en 200 lignes. Cet environnement est très différent de celui de l'EGA par la nature du contrôleur vidéo et par la cartographie mémoire. De plus, en mode émulation CGA, le ROS permet l'appel de l'interruption vidéo 16. Les vecteurs d'interruption correspondant sont donc initialisés par la ROM BIOS IGA de façon à ce que l'on puisse utiliser l'appel ROS standard.

En mode émulation CGA, l'on dispose de deux registres programmables qui définissent le mode de fonctionnement du contrôleur vidéo et la sélection des couleurs. Il s'agit des registres CGA Mode Control et CGA Color Select.

1.11.10.1 Le registre CGA Mode Control

Le registre CGA Mode Control est un registre à écriture seulement auquel on accède par l'adresse d'E/S 3D8h. Il sert à contrôler l'état des circuits vidéo, la sélection du mode graphique ou texte et d'autres sous-fonctions disponibles dans chacun de ces modes.

La composition du registre CGA Mode Control est la suivante :

Bit	Utilisation en sortie
7	Indifférent
6	Indifférent
5	Autorise le clignotement des caractères (inhibe la surbrillance du fond)
4	Sélectionne le mode graphique 2 (dé-sélectionne le mode graphique 1)
3	Active l'affichage vidéo
2	Sélectionne la palette 2 (Dé-sélectionne les palettes 0 et 1)
1	Sélectionne les modes graphiques (dé-sélectionne les modes textes)
0	Sélectionne le mode texte 80 col. (dé-sélectionne le mode texte 40 col.)

Quand le bit 5 est mis à 1 en mode texte, l'avant-plan clignote pour tous les caractères affichés dont le bit d'attribut 7 est à 1 aussi. Le bit 5 n'a aucun effet en mode graphique.

Le bit 4 (Sélection du mode graphique 2) n'a aucun effet en mode texte.

Mettre le bit 3 à 0 entraîne un effacement de l'affichage (en fait un affichage où la couleur d'avant-plan est celle du fond) si le bit 2 du registre IGA Extended Mode Control est à 0 lui aussi.

Le processus standard d'initialisation du mode émulation CGA porte le bit 2 du registre IGA Extended Mode Control à 1 de sorte que l'effacement est inhibé. En affectant le paramètre CGAB à la commande DISPLAY, on peut ré-activer la fonction standard d'effacement.

Le bit 2 (Sélection de la palette 2) n'a pas d'effet en mode graphique 2 ou en mode texte. Il sert en conjonction avec le bit 5 du registre CGA Color Select, à contrôler la palette du mode graphique 1. Pour activer la palette 2, le bit 5 du registre CGA Color Select doit être à 0 et le bit 2 du registre CGA Mode Control à 1.

Le bit 0 (Sélection du mode texte 80 colonnes) n'a aucun effet dans les modes graphiques.

Pour éviter des effets indésirables à l'écran, ce registre doit être mis à jour durant le retour. Tout changement de mode devrait d'ailleurs toujours se faire après avoir désactivé la vidéo. Un changement de mode implique la manipulation des bits 1 et 0 ainsi que, généralement, une reprogrammation du CRTC.

1.11.10.2 Le registre CGA Color Select

Le registre CGA Color Select à écriture seulement est accessible par l'adresse d'E/S 03D9h et sert au contrôle de la couleur du bord (d'écran) en mode texte et à la sélection de la palette, des options de couleur du bord et des pixels en mode graphique. La composition du registre CGA Color Select est la suivante :

Bit	Mode Texte	Graphique 320x200	Graphique 640x200
7,6	Indifférents	Indifférents	Indifférents
5	Indifférent	Sélection Palette 1 Dé-selec. Palette 2	Indifférent
4	Indifférent	Intensité avant-plan pour palettes 0,1,2	Indifférent
3	Intensité (Bord)	Intensité (Fond et Bord)	Intensité (Pixel)
2	Rouge (Bord)	Rouge (Fond & Bord)	Rouge (Pixel)
1	Vert (Bord)	Vert (Fond et Bord)	Vert (Pixel)
0	Bleu (Bord)	Bleu (Fond et Bord)	Bleu (Pixel)

En mode graphique 640x200 avec 2 couleurs, la palette globale de l'écran est contrôlée par la manipulation du registre CGA Color Select (qui est initialisé à 07h par le ROS lors de la sélection du mode 6). Cela signifie que la couleur de l'avant-plan est l'une des 16 couleurs disponibles sur un fond noir. La plupart des applications gèrent le mode graphique 640x200 comme un mode graphique noir et blanc (NB).

Pour éviter des effets d'écran indésirables, ce registre doit être mis à jour durant le retour.

1.11.10.3 Le registre CGA Status

Ce registre d'état du mode CGA est accessible en lecture seule par l'adresse d'E/S 03DA. On peut le lire à tout moment afin d'en retirer les informations suivantes :

Bit	Utilisation en entrée
7-4	Inutilisés
3	Retour de trame (VSYNC)
2	Commutateur du crayon optique ON
1	Latch du crayon optique mis à "VRAI"
0	Activation de l'affichage

Le retour de trame commence lorsque le bord inférieur a été atteint et dure 46 périodes de balayage horizontal (scan) pour se terminer 16 périodes de balayage avant la fin du bord supérieur suivant.

Le bit 2 reflète l'état du bouton poussoir du crayon optique. Il est à 0 quand le bouton est fermé et à 1 quand il est ouvert.

Lorsque le bit 1 est à 1, il indique que le latch (verrou) du crayon optique est activé soit pour avoir détecté le crayon, soit parce qu'on a écrit une donnée vers le canal du crayon optique. Toute écriture de donnée vers le canal de remise à zéro du crayon (Clear Light Pen channel) efface le latch.

Le bit 0 vaut 0 lorsque le signal Vidéo est inhibé et il vaut 1 lorsque le signal de retraçage vertical ou horizontal est actif. Il s'agit là d'une indication en temps réel de l'état de la ligne de balayage dans l'affichage d'une ligne d'écran.

1.11.10.4 L'émulation du contrôleur d'écran 6845 en mode CGA

Le contrôleur CRT (CRTC pour Cathode Ray Tube Controller) est une émulation du contrôleur d'écran MC6845. L'AMSTRAD PC 1640 permet la programmation de tous les registres du 6845 et permet donc une émulation réelle.

Les registres doivent être programmés selon le mode de fonctionnement CGA nécessaire ce qui implique la programmation des registres CGA Mode et Color Select. Un changement de mode doit donc se faire selon la séquence suivante : inhibition de l'affichage, reprogrammation du CRTC avec les paramètres appropriés, reprogrammation des registres Mode et Color Select (la vidéo étant toujours inactive), initialisation de la RAM vidéo en fonction du mode et, enfin, ré-activation de l'affichage.

On contrôle le CRTC au moyen de deux adresses d'E/S correspondant au registre CRTC Address et aux registres de données CRTC.

Le registre CRTC Address (adressage du contrôleur d'écran) est accessible en écriture seule par l'adresse d'E/S 3D4. Ses 5 bits de poids faible permettent d'adresser les registre CRTC Data (registres de données) à l'adresse d'E/S 3D5. Une valeur supérieure à 17 (11h) sera ignorée.

Les registres CRTC sont initialisés comme suit :

N° de regis.	Nom du registre	Mode Texte 40 col. (décimal)	Mode Texte 80 col. (décimal)	Modes graphiques décimal	Lecture (L)/ Ecriture (E)
00	Total Horizontal (-1)	56	113	56	E
01	Affichage Horizontal	40	80	40	E
02	Départ SYNC Horiz. (-1)	45	90	45	E
03	Largeur Synch. Horiz. (-1)	10	10	10	E
04	Total SYNC Verticale (-1)	31	31	127	E
05	Ajustement Total Vertical	6	6	6	E
06	Affichage Vertical	25	25	100	E
07	Départ SYNC Vert. (-1)	28	28	112	E
08	Interlaçage	2	2	2	E
09	Position max. quadrillage (-1)	7	7	1	E
0A	Départ curseur	6	6	6	E
0B	Fin curseur	7	7	7	E
0C	Adresse départ haut	0	0	0	E
0D	Adresse départ bas	0	0	0	E
0E	Position curseur haut	0	0	0	LE
0F	Position curseur bas	0	0	0	LE
10	Position stylo optique haut	-	-	-	L
11	Position stylo optique bas	-	-	-	L

L veut dire Lecture seulement, E écriture seulement et LE Lecture Ecriture.

Une valeur supérieure à 31 dans le registre 0Ah inhibe l'affichage du curseur (parce que le bit 5 est à 1). Le bit 6 de 0Ah sélectionne la fréquence de clignotement.

1.11.10.5 Adressage de l'affichage CRTC

Dans les modes texte, les valeurs de registre pour l'adresse de départ et la position du crayon optique sont dans une fourchette de 8 K allant de 0000h à 1FFFh. La valeur de registre correspondant à une position de caractère dans la RAM d'affichage doit être déterminée en prenant l'adresse paire (de ce caractère) comprise dans la zone de 16 K entre B8000h et BBFFFh, en soustrayant B8000h et en divisant par deux ce résultat.

Dans les modes graphiques, les valeurs de registre pour l'adresse de départ et la position du crayon optique sont dans une fourchette de 4 K allant de 0000h à 0FFFh. Une valeur supérieure à ce dernier chiffre va être prise modulo 0FFFh. Une valeur de registre correspond à deux paires d'octets de la RAM vidéo formant des pixels. Ces octets doivent être pris par mots ; une paire d'octets est affichée sur une ligne de pixels paire et l'autre paire d'octets est affichée sur la ligne impaire suivante à la même position horizontale.

La valeur de registre correspondant à la position d'une paire d'octets (donnant naissance aux pixels) sans la RAM vidéo se calcule ainsi : on prend une adresse paire comprise entre B8000 et B9FFF (soit 8 K). On lui soustrait B8000 et on divise par deux. De la même manière, pour une ligne de pixels impaire, l'offset BA000h est soustrait à une adresse paire comprise entre BA000h et BBFFFh, puis on divise le résultat par 2.

1.11.11 Les registres du mode compatible MDA

Lorsque le PC 1640 IGA se trouve en mode compatible 6845 et dispose d'un écran PC-MD, il peut fonctionner en mode texte MDA (Monochrome Display Adapter). La programmation du contrôleur vidéo (CRTC) et la cartographie mémoire sont très différentes de celles du mode EGA. L'environnement logiciel l'est aussi car la ROM ROS fournit les appels à l'ancienne interruption VIDEO 16. Le BIOS IGA initialise l'IGA en mode émulation MDA et implante les vecteurs de l'interruption 16 de sorte que les appels standard du ROS seront disponibles en mode MDA.

1.11.11.1 Le registre MDA Mode Control

Le registre de contrôle du mode MDA est un registre en écriture seule situé à l'adresse d'E/S 3B8h. Il détermine l'état des circuits vidéo.

Le format du registre MDA Mode Control est le suivant :

Bit	Utilisation en sortie
7	Indifférent
6	Indifférent
5	Active le clignotement (désactive la surbrillance du fond)
4	Indifférent
3	Active l'affichage
2	Indifférent
1	Indifférent
0	Indifférent

Quand le bit 5 est mis à 1, l'avant-plan de tous les caractères affichés, dont le bit 7 de leur octet d'attribut est à 1, clignote.

Le bit 3 doit être mis à 1 pour activer la sortie vidéo et lorsqu'il est à 0, l'affichage est effacé.

1.11.11.2 Le registre MDA Status

Le registre d'état MDA est accessible en lecture seule par l'adresse d'E/S 03BAh. Il donne accès aux informations suivantes :

Bit	Lecture
7-4	Inutilisés
3	Vidéo Mono
2-1	Inutilisés
0	Activation de l'affichage

Le bit 0=0 quand l'affichage est inactif et il vaut 1 lorsque le signal de retraçage vertical ou horizontal est actif.

Le bit 3=1 lorsque la broche (7) de sortie vidéo mono est active. Il s'agit là d'une indication en temps réel de l'état de la ligne de balayage dans l'affichage d'une ligne d'écran.

1.11.11.3 L'émulation du CRTC 6845

Le contrôleur vidéo est une émulation du contrôleur d'écran MC6845. Tous les registres du 6845 sont programmables et les fonctions sont les mêmes que celles du véritable composant.

On contrôle le CRTC au moyen de deux adresses d'E/S correspondant au registre CRTC Address et aux registres de données CRTC.

Le registre CRTC Address (adressage du contrôleur d'écran) est accessible en écriture seule par l'adresse d'E/S 3B4. Ses 5 bits de poids faible permettent d'adresser les registres CRTC Data (registres de données) à l'adresse d'E/S 3B5. Une valeur supérieure à 17 (11h) sera ignorée.

Le nom et les valeurs des registres CRTC sont :

Numéro de regis. (Hex)	Nom du registre	Valeur initiale (décimal)	Type Lecture/ Ecriture
00	Total Horizontal (-1)	97	E
01	Affichage Horizontal	80	E
02	Départ SYNC Horiz. (-1)	82	E
03	Largeur Synch. Horiz. (-1)	15	E
04	Total Vertical (-1)	25	E
05	Ajustement Total Vertical	6	E
06	Affichage Vertical	25	E
07	Départ SYNC Vert. (-1)	25	E

Numéro de registre	Nom du registre	Valeur initiale (décimal)	Type Lecture/ Ecriture
08	Interlaçage	2	E
09	Balayages par ligne (-1)	13	E
0A	Départ curseur	11	E
0B	Arrivée curseur	12	E
0C	Adresse départ haut	0	E
0D	Adresse départ bas	0	E
0E	Position curseur haut	0	LE
0F	Position curseur bas	0	LE
10	Position crayon opt. haut	-	L
11	Position crayon opt. bas	-	L

Mettre le bit 5 du registre 0Ah à 1 inhibe l'affichage du curseur. Le bit 6 de 0Ah sélectionne la seconde fréquence de clignotement du curseur.

1.11.12 L'émulation compatible Hercules

Le mode émultation Monochrome Hercules (HMGA) ressemble fortement en mode texte au mode d'émulation MDA et les registres d'état et de contrôle ont la même disposition. Il y a cependant un certain nombre de fonctions supplémentaires accessibles en émulation HMGA que nous allons examiner.

1.11.12.1 Le registre HMGA Mode Control

Le registre de contrôle du mode émulation Hercules est accessible en écriture seulement à l'adresse d'E/S 3B8h. Il permet de contrôler l'état des circuits vidéo.

Le format du registre HMGA Mode Control est le suivant :

Bit	Utilisation en sortie
7	Sélection de la page d'écran
6	Indifférent
5	Active le clignotement des caractères (désactive la surbrillance du fond)
4	Indifférent
3	Activation de la vidéo
2	Indifférent
1	Sélection modes Texte/Graphique
0	Indifférent

Si le bit 7=0, on sélectionne la page d'écran 0 (B0000-B7FFF) qui est la page choisie par défaut à la mise en route. Si le bit 7=1, cela active la page d'écran 1 (B8000-BFFFF). Notez que cela viendra écraser le tampon d'affichage d'une carte CGA secondaire qui est implanté en B8000-BBFFF. Le bit 1 du registre Hercules Mode (adresse d'E/S 3BFh) interdit lorsqu'il est à 1 le passage à 1 du bit 7 de ce registre.

Lorsque le bit 5 est à 1, l'avant-plan de tous les caractères affichés dont le bit de clignotement (7) de leur octet d'attribut est à 1 clignotera.

Le bit 3 doit être à 1 pour permettre l'affichage et lorsqu'il passe à 0, l'écran est effacé.

Si le bit 1=0, le mode texte est validé. C'est le mode par défaut à la mise sous tension. Mettre le bit 1 à 1 sélectionne le mode graphique. On peut interdire le passage à 1 de ce bit par le bit 0 du registre Hercules Mode (3BFh).

En mode texte, le tampon d'affichage s'étend de B8000 à B0FFF et le ROS du PC 1640 offre 4 pages d'écran dont l'adresse de début est définie en incrémentant l'adresse d'origine de 4 K. Voir le mode BIOS 7 en section 1.11.7. L'organisation des paires caractère/attribut est la même qu'en mode Monochrome Texte.

En mode graphique, on dispose de 2 pages de 32 K sélectionnées par le bit 7 (voir plus haut). La cartographie octet/pixel est segmentée en Quatre groupes de lignes de pixels dans le buffer d'affichage avec une fourchette de coordonnées horizontales de 0 à 719 et une fourchette de coordonnées verticales de 0 à 347. Le programme suivant en C illustre la répartition mémoire :

```
/* ---------------  CARTOHER.C ---------------------- */

#include <stdio.h>

int HGR_Offset(x,y)


{
return(0x2000*(y%4))+(90*(y/4))+(x/8);
}

main()
{
int x,y;

for(x = y = 0 ; y < 10 ; y++)
printf("x = %1d, y = %3d: Offset = %4xh\r\n1",x,y,HGR_Offset(x,y));

for(y = 338 ; y < 348 ; y++)
printf("x = %1d, y = %3d: Offset = %4xh\r\n1",x,y,HGR_Offset(x,y));
}
/* ------------------------------------------------- */
```

Et ce programme donne le résultat suivant :

```
x = 0, y =     0: Offset = 0000h
x = 0, y =     1: Offset = 2000h
x = 0, y =     2: Offset = 4000h
x = 0, y =     3: Offset = 6000h
x = 0, y =     4: Offset = 005Ah
x = 0, y =     5: Offset = 205Ah
x = 0, y =     6: Offset = 405Ah
x = 0, y =     7: Offset = 605Ah
```

```
x = 0, y =     8: Offset = 00B4h
x = 0, y =     9: Offset = 20B4h
x = 0, y =   338: Offset = 5D88h
x = 0, y =   339: Offset = 7D88h
x = 0, y =   340: Offset = 1DE2h
x = 0, y =   341: Offset = 3DE2h
x = 0, y =   342: Offset = 5DE2h
x = 0, y =   343: Offset = 7DE2h
x = 0, y =   344: Offset = 1E3Ch
x = 0, y =   345: Offset = 3E3Ch
x = 0, y =   346: Offset = 5E3Ch
x = 0, y =   347: Offset = 7E3Ch
```

Cet exemple d'exécution concerne les 10 premières et les 10 dernières lignes d'affichage dans le tampon de régénération. Chaque ligne de 720 pixels occupe 90 (5Ah) octets, et toutes les lignes multiples de 4 commencent à la série suivante de 90 octets. Dans le tampon, les quatre groupes sont séparés par 2000h (comme l'illustrent les 4 premières lignes de l'exemple ci-dessus). La ligne finale de chaque groupe est à un offset de 1E95h par rapport à l'origine du groupe. L'espace inoccupé (octet 016Ah) après chaque ligne n'est pas utilisé.

Notez l'expression "+(x/8)" dans l'équation de HGR_Offset. Elle est significative du fait qu'il y a 8 pixels par octet et qu'ils sont numérotés de 0 à 7 à l'inverse de l'ordre habituel de numérotation des bits. L'équation permettant de retrouver le numéro de bit d'un pixel à l'intérieur d'un octet est :

N° de bit = (7-(x Mod 8)).

1.11.12.2 Le registre Status HMGA

Le registre d'état du mode émulation Hercules est un registre à lecture seule auquel on accède par l'adresse d'E/S 03BAh.

Sa lecture peut apporter à tout moment les indications suivantes :

Bit	Utilisation en entrée
7	-SYNC Verticale
6-4	Inutilisés
3	-Vidéo Mono
2-1	Inutilisés
0	Affichage activé

Le bit 7 indique que le signal de synchronisation verticale est actif quand il est à 0 et que l'affichage est inhibé. Il est donc intéressant d'attendre que ce bit soit à 0 pour mettre à jour l'écran. Lorsque le bit 7 est à 1, l'affichage est actif.

Le bit 3=0 quand la broche de sortie vidéo mono est activée (broche 7). Il s'agit d'une indication en temps réel de l'état de la ligne de balayage.

Le bit 0=0 quand le signal vidéo est activé et à 1 quand un signal de retraçage vertical ou horizontal est activé.

1.11.12.3 Emulation du CRTC 6845

Dans ce mode, le contrôleur vidéo est une émulation du contrôleur d'écran MC6845. Tous les 6845 sont programmables dans l'Amstrad PC 1640 et fonctionnent comme le véritable composant.

On contrôle le CRTC au moyen de deux adresses d'E/S correspondant au registre CRTC Address et aux registres de données CRTC.

Le registre CRTC Address (adressage du contrôleur d'écran) est accessible en écriture seule par l'adresse d'E/S 3B4. Ses 5 bits de poids faible permettent d'adresser les registre CRTC Data (registres de données) à l'adresse d'E/S 3B5. Une valeur supérieure à 17 (11h) sera ignorée.

Le nom et les valeurs initiales des registres CRTC sont :

N° de regis	Nom du registre	Mode Texte 80 col. (déc.)	Mode Graph. (déc.)	Type Lect/ Ecrit
00	Total Horizontal (-1)	97	53	E
01	Affichage Horizontal	80	45	E
02	Départ SYNC Horiz. (-1)	82	46	E
03	Largeur Synch. Horiz. (-1)	15	7	E
04	Total Vertical (-1)	25	91	E
05	Ajustement Total Vertical	6	2	E
06	Affichage Vertical	25	87	E
07	Départ SYNC Vert. (-1)	25	87	E
08	Interlaçage	2	2	E
09	Balayages par ligne (-1)	13	3	E
0A	Départ curseur	11	0	E
0B	Arrivée curseur	12	0	E
0C	Adresse départ haut	0	0	E
0D	Adresse départ bas	0	0	E
0E	Position curseur haut	0	0	LE
0F	Position curseur bas	0	0	LE
10	Posit. crayon opt. haut	-	-	L
11	Posit. crayon opt. bas	-	-	L

Mettre le bit 5 du registre 0Ah à 1 inhibe l'affichage du curseur. Le bit 6 de 0Ah sélectionne la seconde fréquence de clignotement du curseur.

La ROM ROS du PC 1640 n'offre que le mode texte et ne contient pas de valeurs d'initalisation pour le mode graphique HMGA. Les valeurs données ici sont cependant des valeurs habituelles et représentent la mise en état de la page 0. Consultez la documentation du MC6845 pour davantage d'information sur d'autres configurations.

1.12 LE CONTROLEUR DE LECTEURS DE DISQUETTES

Le contrôleur de lecteurs de disquettes (FDC pour Floppy Disk Controller) est bâti autour du uPD765A de NEC et permet le contrôle d'un ou de deux lecteurs 5 pouces un quart en simple ou double face. Son mode de fonctionnement est à modulation de fréquence modifiée (MFM) et les disquettes sont formatées en double densité. La vitesse de transfert est de 250 kilobits par seconde.

Le FDC se programme par le biais du registre Drive Selection à l'adresse d'E/S 3F2. Il est défini comme suit :

Bit (Dn)	Utilisation en sortie
7-6	Inutilisés
5	Mise en route moteur(s) et permet la sélection du lect. 1
4	Mise en route moteur(s) et permet la sélection du lect. 0
3	Autorise les demandes d'interruption et de DMA
2	-reset du 765A
1	Bit 1 de sélection de lecteur (DS1)
0	Bit 0 de sélection de lecteur (DS0)

Les bits de sélection de lecteur (DS1, DS0) ne sont valides qu'avec les valeurs 00 et 01 pour, respectivement, le lecteur 0 ou 1. La sélection réelle du lecteur n'est effectuée que quand le bit 4 pour le lecteur 0 ou le bit 5 pour le lecteur 1 est activé (mis à 1). Réciproquement, le passage à 1 de l'un des bits 4 ou 5 ne sera validé que par le passage à 1 du bit DS correspondant.

Le bit 2 (-reset 765A) doit être forcé au niveau bas (0) et maintenu bas durant au moins 3,5 uS pour que le reset du 765A soit effectif. Il doit être remis à 1 pour stopper le signal de reset.

A la mise sous tension ou après un reset système, tous les bits de ce registre sont remis à 0.

1.12.1 Environnement Hardware du FDC

Le matériel impose les conditions d'installation suivantes à l'utilisation du contrôleur et des lecteurs de disquette :

1. La fréquence d'horloge du contrôleur 765A est fixée à 4,0 MHz.

2. Les transferts de données se font en accès mémoire direct (DMA) en utilisant le contrôleur DMA de la carte mère. La sortie DRQ du 765A peut être connectée ou déconnectée de l'entrée DRQ2 du contrôleur DMA par logiciel en se servant du bit 3 du registre Drive Selection.

3. Le 765 dispose d'un niveau d'interruption pour signaler la fin d'exécution d'une commande et faire passer des messages d'avertissement d'état au CPU. La sortie INT du 765A peut être connectée ou déconnectée par voie logicielle de l'entrée IRQ6 du contrôleur d'interruptions. On se sert pour cela du bit 3 du registre Drive Selection.

4. Le lecteur 0 est toujours présent, le lecteur 1 est optionnel. La connexion des lecteurs 2 et 3 n'est pas implémentée, on ne peut accéder à eux (il n'y a pas de décodage de signal de sélection). Le signal de sortie Drive Ready (lecteur prêt) du lecteur courant sélectionné est relié à l'entrée RDY du 765A. Pour les lecteurs qui ne disposeraient pas d'un tel signal, on peut éventuellement fixer RDY sur la condition VRAI.

5. Les sorties Drive Select du 765A, US1 et US0, ne sont pas utilisées pour la sélection du lecteur. Cette fonction est assurée en même temps que le contrôle du moteur par le registre Drive Selection qui est un élément externe au FDC 765A.

6. L'entrée FLT (Fault : erreur) du 765A est forcée à l'état FAUX en permanence.

7. Un signal d'état Two-Sided (double face) depuis les lecteurs n'est pas prévu mais l'interfaçage aux lecteurs permet leur utilisation en double face.

8. Une précompensation en écriture de 250 uS est effectuée.

9. On peut réinitialiser individuellement le 765A par logiciel au moyen du bit 2 du registre Drive Selection.

1.13 LE PORT SERIE ASYNCHRONE RS232C

Le port série asynchrone est construit autour du composant NATIONAL INS 8250 ACE (UART) à un canal unique.

La fréquence d'horloge à l'entrée du 8250 est à 1,8432 MHz (+/- 0,1%).

La sortie BAUD OUT du 8250 est connectée à l'entrée RCLK.

Le 8250 peut utiliser un niveau d'interruption. Lorsque la sortie OUT2 du 8250 est amenée au niveau bas (c'est-à-dire quand on écrit un 1 dans le bit 2 du registre MODEM Control du 8250), le signal INTRPT est envoyé à l'entrée IRQ4 du contrôleur d'interruption.

1.13.1 L'interface série

L'interface série utilise le connecteur subminiature 25 voies de type DB25 mâle émulant un DTE (Data Terminal Equipment).

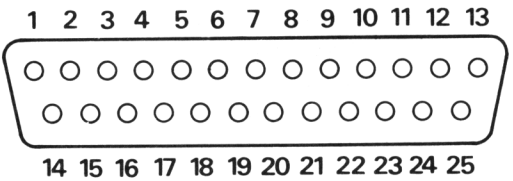
Les niveaux éléctriques des signaux sur cette interface satisfont à la norme RS-232C de l'EIA (Electronics Industry Association) et au standard d'interfaçage CCITT V.24.

Les signaux de contrôle et de réception entre le connecteur et le 8250 sont tous en inversion.

1.13.2 Brochage du connecteur série

Broche	EIA	CCITT	Description
2	BA	103	TxD Sortie données série
3	BB	104	RxD Entrée données série
* 4	CA	105	RTS Demande d'émission Sortie
5	CB	106	CTS Accord réception Entrée
6	CC	107	DSR Emission prête Entrée
7	AB	102	Masse des signaux (retour commun)
8	CF	109	DCD Détection de porteuse
* 20	CD	108.2	DTR Terminal prêt Sortie
22	DE	125	RI Flag de sonnerie Entrée

* Ces signaux de contrôle, quand ils sont présents, doivent permettre de détecter un mise hors tension du dispositif à l'autre bout de l'interface (la ligne) ou la déconnection du câble de liaison. Ces conditions doivent être interprétées comme une fin de connexion.



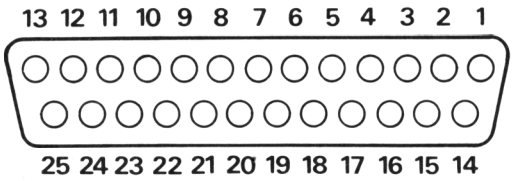
Reportez-vous à l'annexe 4 pour une description plus détaillée des principes de connexion RS232.

1.14 L'INTERFACE PARALLELE D'IMPRIMANTE

Le port parallèle d'imprimante est décrit dans la section 1.1. Il s'agit d'une interface 8 bits de type Centronics. L'interface imprimante utilise un connecteur subminiature de type D25 (femelle) situé à l'arrière du PC 1640.

Le brochage de ce connecteur est le suivant :

Broche	Fonction
1	Data Strobe Off
2	Bit 0 des Données (poids faible)
3	Bit 1 des Données
4	Bit 2 des Données
5	Bit 3 des Données
6	Bit 4 des Données
7	Bit 5 des Données
8	Bit 6 des Données
9	Bit 7 des Données
10	0 = Accusé de réception de l'imprimante
11	Printer Buzy, 1 = Imprimante occupée
12	Paper Out, 1 = Abscence de papier
13	Select Printer, Sélection d'imprimante
14	Not Select Auto Feed, 0 = Saut de ligne auto
15	Not Printer Error, 0 = Erreur imprimante
16	Not Reset Printer, 0 = Reset imprmante
17	Not Printer Selected, 0 = Imprimante sélectionnée
18	GND Masse (électrique)
19	GND Masse (électrique)
20	GND Masse (électrique)
21	GND Masse (électrique)
22	GND Masse (électrique)
23	GND Masse (électrique)
24	GND Masse (électrique)
25	GND Masse (électrique)



L'annexe 5 contient les spécifications de connexion avec l'imprimante DMP 3000.

1.15 L'INTERFACE CLAVIER

Les données provenant du clavier sont reçues par le CPU grâce au port A de l'interface parallèle programmable (PPI) 8255 et l'interruption clavier (niveau 1) du contrôleur d'interruptions programmable (PIC) 8259A. L'un et l'autre ont été décrits préalablement (1.6 et 1.8).

1.15.1 Les signaux Serial Clock et Serial Data

Les signaux Serial Clock (horloge série) et Serial Data (données série) servent à l'interfaçage du clavier. Ils sont tous les deux bi-directionnels et permettent au microcontrôleur du clavier d'envoyer des codes de touches à la carte mère qui elle-même se sert de ces deux signaux pour indiquer qu'elle est prête à recevoir un autre code. De plus, ces signaux servent à commander un reset du microcontrôleur clavier soit par voie hardware soit par voie logicielle.

1.15.2 Interface Clavier vers Unité Centrale

Au repos, les signaux Serial Clock et Serial Data sont au niveau haut. Il faut au moins un délai de 5 uS entre le passage d'un état à l'autre de l'un de ces signaux ou le passage d'un signal à l'autre.

Les codes de touches sont envoyés depuis le contrôleur du clavier jusqu'à l'unité centrale en mode série 8 bits, le bit de poids fort en tête. Les données de codes de touches provenant du clavier sont décodées grâce à une base de temps dans un registre de décalage (shift register) comme représentant un 1 ou un 0. Pour être interprété comme un bit valant 1, le signal Serial Data doit rester au niveau haut durant le temps que met Serial Clock à passer au niveau bas et à revenir au niveau haut. Pour que l'on ait un bit valant 0, Serial Data doit se trouver au niveau bas avant le passage de Serial Clock au niveau bas. Serial Data repasse au niveau haut suivi par Serial Clock. Les bits 1 ou 0 sont mesurés par le registre de décalage au moment du front descendant du signal Serial Clock.

1.15.3 Unité Centrale vers Interface Clavier

Quand elle reçoit un code clavier du microcontrôleur, dans les 5 uS suivant la dernière chute de Serial Clock, l'électronique de la carte-mère force Signal Data au niveau bas et l'y maintient jusqu'à ce qu'elle soit prête à recevoir un nouveau code de touche. Dès que l'unité centrale a libéré Serial Data (au niveau haut), le clavier est libre d'envoyer de nouvelles données. Cette réponse à la réception d'un code touche est appelée une séquence d'ACKNOWLEDGE (acquitement ou accusé de réception).

L'unité centrale commande un reset au microcontrôleur du clavier en forçant la ligne Serial Clock au niveau bas pendant au moins 10 millisecondes. L'état de Serial Data n'affecte pas cette opération.

1.15.4 Les codes de touches (Keycodes)

Les données du clavier s'exprimant sur 8 bits, on dispose donc de 128 codes d'établissement de touche (quand on appuie dessus) et de 128 codes de rupture de touches (quand on les relâche). Quel que soit la touche, le code d'établissement est compris entre 0 et 127 (décimal). Le code de rupture d'une touche est le même que son code d'établissement mais avec le bit de poids le plus fort mis à 1 ; on obtient donc une fourchette allant de 128 à 255.

Après que l'on ait appuyé sur une touche et que son code ait été envoyé à l'unité centrale, si la pression sur la touche est maintenue plus d'une seconde et que l'on n'appuie pas sur une autre touche, le microcontrôleur envoie de nouveau le code toutes les 83 millisecondes à condition que l'unité centrale a fait savoir qu'elle était prête à le recevoir.

Le code de touche 0AAh est envoyé après un reset pour indiquer le déroulement avec succès des autotests de mise en route.

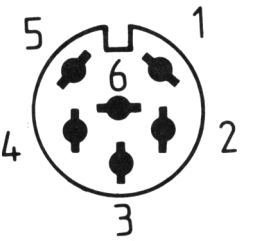
La ROM BIOS du PC 1640 reçoit les codes de touches via une sous-routine d'interruption et elle les codent en une valeur sur 2 octets (un token, que nous appelons code clavier dans cet ouvrage pour les différencier des codes caractères ASCII) qui sera ensuite transformée en codes ASCII par les applications. Les valeurs de codes de touches et les codes clavier correspondants sont détaillés dans la section 2.3.5. Des programmes tels que KEYBFR.EXE remplacent les vecteurs de la ROM BIOS ayant trait à l'encodage des codes de touches et réalisent leurs propres traitements. Etant donné que certains codes de touches sont en fait des codes relatifs aux boutons de la souris et au joystick, l'utilisation d'un KEYBxx ne provenant pas d'AMSTRAD fera perdre les codes spéciaux correspondants et les fonctionnalités qui leur sont attachées.

1.15.5 Le connecteur du clavier

Le connecteur du clavier est un connecteur à 6 broches de type DIN. L'attribution des broches est la suivante :

Broche	Fonction	Explication
1	KBCLK	Serial Clock
2	KBDATA	Serial Data
3	M1	Bouton 1 de la souris
4	Masse	
5	+5 V continu	
6	M2	Bouton 2 de la souris

Vue du côté gauche de la machine



Les signaux M1 et M2 proviennent directement du connecteur de la souris pour aller vers le contrôleur du clavier qui produira les codes touches correspondants.

1.16 L'INTERFACE SOURIS

L'interface de la souris doit traiter deux entrées correspondant à l'état des boutons et deux valeurs sur 8 bits contrôlant le mouvement dans les plans X et Y. Les commutations des deux boutons M1 et M2 sont intégrées dans la matrice du clavier et traitées comme des codes de touches (produisant repectivement les codes d'établissement 7E et 7D).

Le registre Mouse X-Coordonate (Coordonnée X de la souris) à l'adresse d'E/S 078h est un compteur 8 bits qui peut être lu par le CPU. Un accès en écriture, quel que soit la valeur, efface le compteur. De la même manière le compteur Mouse Y-Coordonate (adresse d'E/S 07Ah) peut être lu par le CPU ou effacé par l'écriture d'une valeur quelconque à son adresse d'E/S.

Les compteurs sont incrémentés ou décrémentés en fonction de la direction du mouvement de la souris et leur valeur indique le déplacement depuis la dernière lecture ou l'effacement. Le compteur X s'incrémente durant um mouvement vers la droite et se décrémente quand la souris va vers la gauche. Le compteur Y est incrémenté pour un mouvement vers le haut et décrémenté pour un déplacement vers le bas. Pour refléter fidèlement les mouvements de la souris, les programmes doivent lire et effacer les compteurs de la souris à une cadence suffisamment soutenue pour prévenir un débordement d'une valeur positive vers une valeur négative ou vice versa lorsque la souris est déplacée à un rythme soutenu. Un mouvement de la souris de 1/8 mm incrémente ou décrémente le compteur correspondant d'une unité.

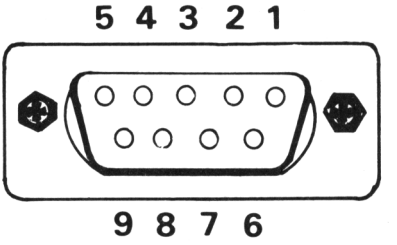
Les systèmes d'exploitation fournis avec cette machine et qui possèdent des drivers particuliers pour la souris Amstrad effectuent un cycle lecture de compteur et effacement toutes les 18 ms grâce à l'interruption ticker. Une routine qui n'irait lire les compteurs qu'occasionnellement risquerait alors de ne remarquer que de faibles mouvements de la souris parce que plusieurs effacements auraient eu lieu entre deux lectures de cette routine. Reportez-vous à l'annexe 1 pour des informations complémentaires sur les interfaces logicielles avec la souris.

1.16.1 Le connecteur souris

Le connecteur de la souris est un connecteur femelle 9 broches de type D situé sur la gauche de la machine et dont le brochage est propre à Amstrad. La connexion de matériels provenant d'autres constructeurs (même avec une prise identique) risque d'endommager à la fois le PC 1640 et l'accessoire inadapté.

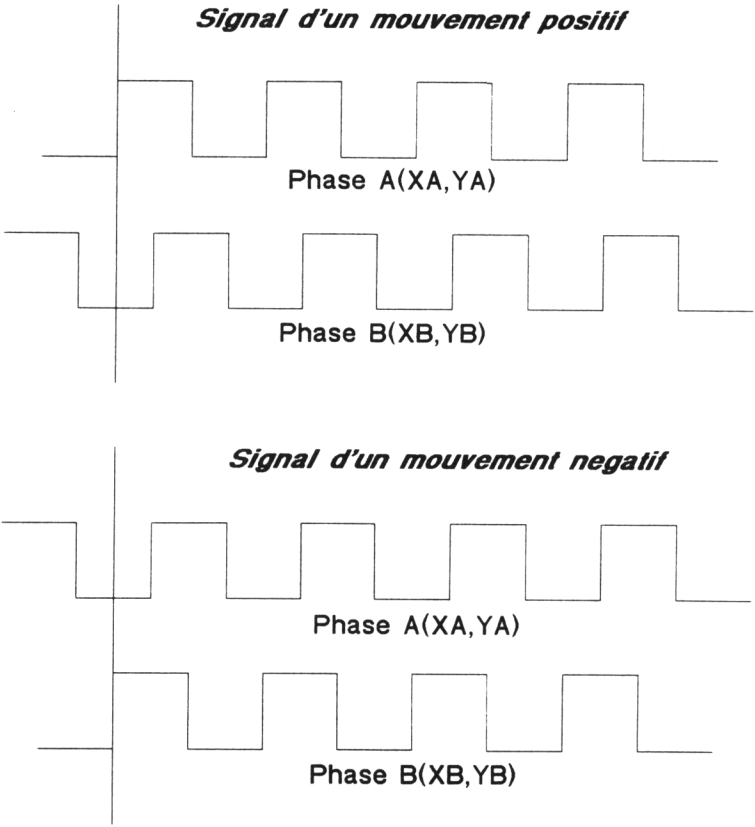
La disposition des broches est la suivante :

Broche	Fonction
1	XA
2	XB
3	YA
4	YB
5	Réservé
6	M1
7	+ 5V continu
8	Masse
9	M2



Connecteur Souris

Les quatre premières broches transportent les ondes carrées XA, XB, YA et YB encodées optiquement. Pour un mouvement positif, l'onde carrée de la phase A précède la phase B de 90 degrés et la situation inverse se produit lors d'un mouvement négatif. Les broches restantes transportent les signaux des boutons 1 et 2 (M1 et M2) de la souris, l'alimentation 5 volts et l'on trouve enfin la masse.



1.17 L'INTERFACE JOYSTICK

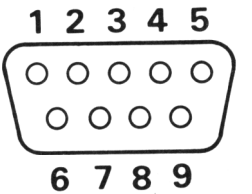
L'AMSTRAD PC comporte une interface joystick conforme au standard industriel. Les entrées du joystick sont traitées comme des codes touches provenant de l'interface clavier. Les codes d'établissement (qui ont un code faible) débutent à 7Ch et descendent jusqu'à 77h et correspondent dans l'ordre à Haut, Bas, Gauche, Droite, Feu1, Feu 2. Le ROS (voir chapitre 2) transforme ces codes de touches d'un octet en codes claviers de deux octets (tokens). On peut attribuer un code clavier déterminé aux deux boutons feux. Ces codes figurent dans la NVR.

1.17.1 Le connecteur du joystick

Il s'agit d'un connecteur à 9 broches de type D (mâle) dont le brochage est au standard industriel. La connexion de matériels inadaptés au PC 1640 (même avec une prise identique) risque d'endommager à la fois l'ordinateur et l'accessoire (joystick).

La prise du joystick est située sur l'arrière de la face gauche du clavier. Son brochage est le suivant :

Broche	Fonction
1	Haut
2	Bas
3	Gauche
4	Droite
5	Réservé
6	Feu2
7	Feu1
8	Masse électrique
9	Non connectée



Connecteur du joystick

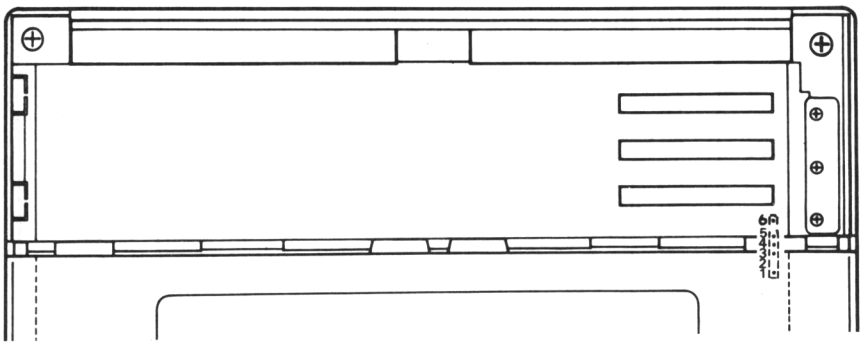
1.18 LE CONNECTEUR DU CRAYON OPTIQUE

L'AMSTRAD PC 1640 fournit une interface standard pour le crayon optique via le MC6845 émulé. On accède au connecteur du crayon optique en ôtant le couvercle des connecteurs d'extention sur le

dessus et à l'arrière de la machine. Le connecteur est situé à l'intérieur de la machine, à droite et à l'extrême bord de la carte principale (juste en deçà des connecteurs d'extension). C'est un connecteur de type berg à 6 broches et il est repéré par un PL8 en grosses lettres. La broche 1 est la broche la plus proche du lecteur de disquette.

Le brochage est le suivant :

Broche	Fonction
1	-Entrée crayon
2	(Keyway)
3	Interrupteur crayon
4	Masse
5	+ 5V DC
6	+ 12V DC

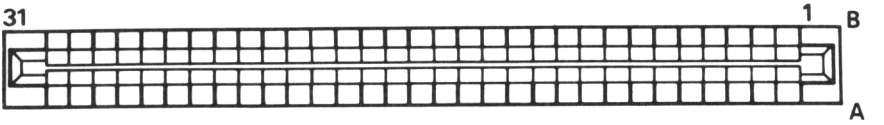


1.19 LES CONNECTEURS D'EXTENSION

L'AMSTRAD PC 1640 possède 3 connecteurs d'extension pour l'ajout de cartes périphériques. Ces connecteurs (slots) sont situés à droite de la machine et à l'arrière de la carte mère. La numérotation des broches de chaque connecteur est identique à celle des autres. Les broches du côté gauche du connecteur (côté passif de la carte d'extension) sont numérotées de haut en bas B1 à B31 et celles du côté droit (face de la carte portant les composants) sont numérotées A1 à A31 toujours de haut en bas. Le tableau suivant donne l'assignation des broches du connecteur :

Broche	Signal	Entrée/Sortie
A01	-I/OCHCK (bas : erreur de parité)	E
A02	D7 (bus de données bit 7)	E/S
A03	D6 (bus de données bit 6)	E/S
A04	D5 (bus de données bit 5)	E/S
A05	D4 (bus de données bit 4)	E/S
A06	D3 (bus de données bit 3)	E/S
A07	D2 (bus de données bit 2)	E/S
A08	D1 (bus de données bit 1)	E/S
A09	D0 (bus de données bit 0)	E/S
A10	I/O RDY	E
A11	AEN (Adressage autorisé)	S
A12	bit A19 (bus d'adresse)	S
A13	bit A18 (bus d'adresse)	S
A14	bit A17 (bus d'adresse)	S
A15	bit A16 (bus d'adresse)	S
A16	bit A15 (bus d'adresse)	S
A17	bit A14 (bus d'adresse)	S
A18	bit A13 (bus d'adresse)	S
A19	bit A12 (bus d'adresse)	S
A20	bit A11 (bus d'adresse)	S
A21	bit A10 (bus d'adresse)	S
A22	bit A09 (bus d'adresse)	S
A23	bit A08 (bus d'adresse)	S
A24	bit A07 (bus d'adresse)	S

A25	bit A06 (bus d'adresse)	S
A26	bit A05 (bus d'adresse)	S
A27	bit A04 (bus d'adresse)	S
A28	bit A03 (bus d'adresse)	S
A29	bit A02 (bus d'adresse)	S
A30	bit A01 (bus d'adresse)	S
A31	bit A00 (bus d'adresse)	S
B01	Masse	-
B02	RESET	S
B03	+5V DC	-
B04	IRQ2 (Demande d'interruption 2)	E
B05	-5V DC	-
B06	DREQ2 (demande de DMA2)	E
B07	-12V DC	-
B08	Réservée	E
B09	+12V DC	-
B10	Masse	-
B11	-MEW (écriture mémoire)	S
B12	-MRD (lecture mémoire)	S
B13	-IOW (écriture d'E/S)	E/S
B14	-IOR (lecture d'E/S)	E/S
B15	-DACK3 (DMA 3 autorisé)	S
B16	DREQ3 (Demande de DMA 3)	E
B17	-DACK1 (DMA 1 autorisé)	S
B18	DREQ1 (Demande de DMA 1)	E
B19	-DACK0 (DMA 0 autorisé)	S
B20	CLK (Horloge)	S
B21	IRQ7 (Demande d'interruption 7)	E
B22	IRQ6 (Demande d'interruption 6)	E
B23	IRQ5 (Demande d'interruption 5)	E
B24	IRQ4 (Demande d'interruption 4)	E
B25	IRQ3 (Demande d'interruption 3)	E
B26	-DACK2 (DMA 2 autorisé)	S
B27	T/C (compte terminal)	S
B28	ALE (autoris. verrouil d'adresse)	S
B29	+5V	-
B30	CK14	S
B31	Masse	-



Connecteur du bus d'extension

Vue de dessus, face à la machine.

Les connecteurs d'extension d'E/S sont conformes au standard industriel en matière de bus d'ordinateur personnel 16 bits. La transition entre le bus interne 16 bits du 8086 et le bus d'E/S externe 8 bits est assurée par les circuits de la carte mère. Un transfert d'E/S sur 16 bits du CPU sera coupé en deux cycles sur 8 bits (avec états d'attente).

Tous les signaux sont compatibles TTL et peuvent supporter au maximum deux low-power schottky (LSTTL) par connecteur. La consommation électrique doit être limitée (par connecteur) à un maximum de 700 milliampères sur le +5V, 100 milliampères sur le -5V, 700 milliampères sur le +12V et 100 milliampères sur le -12V.

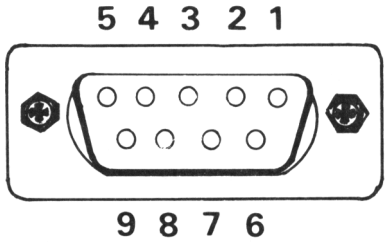
Notez que l'accès direct sur le bus mémoire rapide 16 bits de la carte mère n'est pas possible depuis les connecteurs d'extension.

Des informations complètes sur la conception des cartes prototypes doivent être fournies avec la documentation de cet équipement.

1.20 LE CONNECTEUR VIDEO

Ce connecteur situé à l'arrière de la machine est de type D (femelle) à 9 broches. Son brochage est le suivant :

Broche	Fonction
1	Masse
2	Rouge secondaire (r) ou masse
3	Rouge primaire (R)
4	Vert primaire (G)
5	Bleu primaire (B)
6	Vert secondaire (g) ou intensité (I)
7	Bleu secondaire (b) ou vidéo mono (V)
8	SYNC horizontale
9	SYNC verticale



Connecteur Vidéo

Le même brochage sert aux trois types d'écrans mais l'interprétation de la fonction d'une broche particulière peut varier d'un modèle à l'autre. De plus, certains équipements s'attendent à ce que la broche 2 soit une masse à part entière, ce qui peut être réalisé grâce au commutateur 8 (sw8) à l'arrière du PC 1640. Quand le switche 8 est sur la position OFF, la broche 2 est à la masse. Quand le sw8 est sur ON, le signal du rouge secondaire passe par la broche 2 (voir la section 1.22).

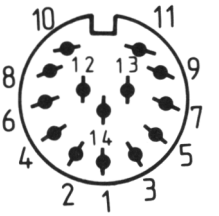
1.21 LE CONNECTEUR D'ALIMENTATION

Le connecteur d'alimentation est de type DIN et possède 14 broches. Il est situé à l'arrière de l'ordinateur.

Le courant provient de l'alimentation située dans le moniteur.

Le brochage du connecteur est le suivant :

Broche	Fonction
1	Non connectée
2	0 Volts continu
3	+5 Volts continu
4	0 Volts continu
5	+5 Volts continu
6	Non connectée
7	-12 Volts continu
8	0 Volts continu
9	Non connectée
10	0 Volts continu
11	+12 Volts continu
12	0 Volts continu
13	-5 Volts continu
14	Non connectée



Connecteur d'alimentation

1.22 CONFIGURATION DES COMMUTATEURS D'AFFICHAGE DU PC 1640

Les commutateurs déterminant le mode graphique par défaut de l'IGA sont ainsi définis :

sw4	sw3	sw2	sw1	Contrôleur principal	Contrôleur secondaire
OFF	ON	OFF	OFF	MD (Mono) Interne	CGA (CO 80) Externe
OFF	ON	OFF	ON	MD (Mono) "	CGA (CO 40) "
OFF	ON	ON	OFF	ECD350 (CO80) "	MDA/HERC (Mono) "
OFF	ON	ON	ON	*ECD200 (CO80) "	MDA/HERC (Mono) "
ON	OFF	OFF	OFF	*CD (CO80) "	MDA/HERC (Mono) "
ON	OFF	OFF	ON	CD (CO40) "	MDA/HERC (Mono) "
ON	OFF	ON	OFF	CGA (CO80) Externe	MD (Mono) Interne
ON	OFF	ON	ON	(CO 40) "	MD (Mono) "
ON	ON	OFF	OFF	MDA/HERC (Mono)"	ECD350 (CO80) "
ON	ON	OFF	ON	MDA/HERC (Mono)"	ECD200 (CO80) "
ON	ON	ON	OFF	MDA/HERC (Mono)"	CD(CO80) "
ON	ON	ON	ON	MDA/HERC (Mono)"	CD(CO40) "

* Voir note 3 plus loin.

sw5 : OFF pour un vrai mode EGA
ON pour les modes émulation CGA/MDA/HERC (aussi appelés modes compatibles 6845)

sw8 : OFF pour les moniteurs CD (RGB standard)
ON pour les moniteurs ECD (RGB haute définition)

Les commutateurs de sélection de jeux de caractères étrangers sont :

sw10	sw9	sw7	sw6	
OFF	OFF	--	--	Jeu anglais
OFF	ON	ON	ON	Jeu anglais
OFF	ON	OFF	OFF	Jeu danois
OFF	ON	OFF	ON	Jeu portugais
OFF	ON	ON	OFF	Jeu grec

Les commutateurs sw6, sw7, et sw10 sont utilisés afin que la ROM BIOS IGA sélectionne un jeu de caractères alterné à charger dans la RAM des fontes IGA (pour les caractères de 128 à 255).

Quand le sw9 est mis sur OFF, cela inhibe la sélection des jeux de caractères étrangers de la ROM IGA (dans la zone C4000h-C7FFFh). Cela force le BIOS IGA à sélectionner les caractères anglais qui sont situés en dessous de C4000. Mettre ce switche 9 sur OFF permet aussi à un dispositif hardware de placer sa ROM contrôleur dans l'espace C4000-C7FFF sans entrer en conflit avec l'implantation des jeux de caractères.

L'IGA peut être désactivé en mettant le sw10 sur OFF. Dans ce cas, les switches 1 à 5, 8 et 9 n'ont plus d'effet et de plus, les commutateurs 6 et 7 assument une fonction différente de celle que leur attribue le ROS PC 1640 pour la définition du mode d'affichage par défaut. Voici ces nouvelles fonctions :

sw10	sw9	sw7	sw6	
ON	--	OFF	OFF	EGA externe installé
ON	--	OFF	ON	CGA externe en mode 40 col.
ON	--	ON	OFF	CGA externe en mode 80 col.
ON	--	ON	ON	Adaptateur monochrome externe (MDA ou Hercules)

Notes :

1. Durant la mise en route de la machine, tous les messages sont affichés sur l'écran principal.

2. Pour envoyer les affichages de texte à l'adaptateur vidéo secondaire, il faut se servir de l'utilitaire MODE. Le tableau des commutateurs sw1-sw4 donne le paramètre approprié pour MODE.

3. Les positions de commutateurs marquées ECD350 et ECD200 indiquent que l'affichage se fera en haute résolution (350 lignes) ou en basse résolution (200lignes). ECD200 est identique à CD (CO80) dans le mode texte par défaut. Cependant les logiciels graphiques peuvent examiner la configuration des switches pour savoir si le mode 350 est accessible. Une mauvaise configuration peut amener le programme à passer en 350 lignes et si un écran PC-ECD n'est pas réellement connecté, la synchronisation sera incorrecte. Bien que cela ne puisse endommager le moniteur, l'écran sera instable et la machine devra être relancée pour y remédier.

4. Les adaptateurs secondaires figurant dans le tableau des switches sw1-sw4 sont optionnels lorsqu'ils sont indiqués comme externes et la configuration des commutateurs informe le logiciel comme le BIOS ou MODE.EXE de ce que seraient les valeurs à prendre en compte si l'écran était installé. Quand l'adaptateur principal est noté comme externe, il est obligatoire et le lancement de la machine ne pourrait avoir lieu s'il n'était pas installé.

CHAPITRE 2

SYSTEME D'EXPLOITATION

Ce chapitre décrit le système d'exploitation résident (ROS) et le BIOS en ROM de l'adaptateur graphique interne (IGA pour Internal Graphics Adapter) de l'Amstrad PC 1640. Il définit les interfaces logicielles de toutes les interruptions de la ROM du ROS Amstrad PC 1640 et toutes les adresses en mémoire vive qu'elle utilise.

Le message de copyright suivant se trouve au début du ROS et commence à l'offset 0003 (par rapport à l'adresse FC000) :

(C) Copyright 1987 Amstrad plc

Physiquement, le ROS occupe les 16 K les plus hauts (de FC000 à FFFFF) de la zone d'adressage de 1 M octets du 8086-2 (voir figure 1.1). La zone de 64 K allant de F0000 à FFFFF est réservée à la ROM système et comprend l'adresse FFFF0 d'initialisation et de reset. Le décodage des adresses du PC 1640 s'effectue de telle manière pour la zone de la ROM système, que le ROS, dont la mémoire morte réside en fait à FC000, est répété quatre fois à partir de F0000.

NOTE : Sauf précision contraire, toutes les constantes d'adresses de ce document sont exprimées en hexadécimal et toutes les autres valeurs en décimal.

Tous les appels au ROS et au BIOS IGA doivent se faire par les interruptions logicielles décrites dans ce manuel. Les programmes d'application ne doivent pas tenter d'accéder directement aux routines en mémoire morte. AMSTRAD se réserve le droit de modifier le firmware composant la ROM du système d'exploitation résident (ROS) et la ROM IGA s'il le juge nécessaire.

Le système d'exploitation comprend un ensemble de routines résidentes assurant plusieurs fonctions utilitaires et d'entrées-sorties :

1. Initialisation à la mise en route et autotest.
2. Entrées clavier.
3. Affichage des caractères et pixels.
4. Dumping écran des tampons vidéo.
5. Traitement des E/S de caractères avec les ports imprimante et série.
6. Manipulation de l'horloge système et de l'horloge temps réel.
7. Opérations d'E/S des lecteurs de disquette en lecture, écriture et formatage.

Nous rappelons que pour assurer la portabilité des programmes d'application, tous les processus d'E/S doivent passer par les interruptions logicielles. Cela permet d'éviter les problèmes occasionnés par une modification ou une amélioration du hardware.

2.1 INITIALISATION ET AUTOTEST

La routine d'initialisation et d'autotest a son point d'entrée en FFFF0 qui est le point d'entrée du reset du CPU. On trouve ici un ensemble de routines qui assurent toutes les initialisations et tous les autotests hardware nécessaires : mise en état de la zone RAM des variables du BIOS, initialisation de toutes les adresses des interruptions utilisées par le ROS, initialisation des ROM des périphériques des connecteurs d'extension (slots) ainsi que le chargement et l'exécution du vecteur d'initialisation (bootstrap) de la disquette ou du disque dur.

Le ROS ne se sert pas de la RAM du système (User Ram Area) pour la pile ou des variables tant que celle-ci n'a pas été testée avec succès. S'il découvre une erreur de RAM, il affiche normalement un message d'avertissement, à condition évidemment qu'il n'y ait pas, d'autre part, un dysfonctionnement du CPU ou des circuits de visualisation.

Les initialisations de mise en route et l'autotest se déroulent comme suit :

1. *Les interruptions masquables et non masquables sont inhibées.*

2. *Exécution de l'autotest qui se décompose ainsi :*

   Checksum du ROS
   Test de toute la RAM du système
   Test du contrôleur DMA 8237
   Test du timer d'intervalle programmable 8253
   Test de l'interface programmable parallèle 8255
   Vérification du comptage de l'horloge temps réel
   Test de l'interface série du système
   Test de l'interface parallèle du système
   Test du contrôleur d'interruptions programmable 8259
   Test des registres compteurs X et Y de la souris

Après un reset, tous les tests sont exécutés de nouveau hormis ceux de la mémoire vive. Si les trois liaisons d'options de poids le plus faible du registre de statut de l'imprimante système sont tous mis à 1, le mode diagnostic est validé et si la ROM de diagnostic est présente, ses routines de tests sont exécutées. Sinon, seuls les tests du clavier et des disques sont lancés avant l'initialisation de la ROM externe.

Reportez-vous à la section 2.2 pour la description détaillée des tests de mise en route.

3. Checksum de la NVR

Si la somme de contrôle de la NVR est incorrecte, celle-ci reçoit alors ses valeurs par défaut. Voir section 2.5.

4. Initialisation du timer d'intervalle programmable 8253

Le compteur 0 est réglé pour une interruption toutes les 54.934 millisecondes. Le compteur 1 est initialisé pour qu'il génère un signal de sortie avec une périodicité de 15.13 microsecondes. Le compteur 2 est invalidé.

5. Initialisation du contrôleur DMA 8237

Le canal DMA 0 est configuré pour le rafraîchissement de la mémoire. Les canaux 1, 2 et 6 sont mis hors service.

6. Initialisation du contrôleur d'interruptions programmable 8259

Invalide (masque) tous les niveaux d'interruption. Notez que les niveaux 0, 1 et 3 seront validés par la suite.

7. Initialisation des registres Write Status

Write Status 1 est initialisé avec un octet de la NVR qui définit le nombre de lecteurs installés et le mode vidéo par défaut. Le ROS manipule aussi le bit 1 du registre d'état en fonction de la présence ou de l'absence d'un coprocesseur 8087. Voir section 1.8.4 pour le détail de la configuration du WSS1.

Write Status 2 est initalisé en fonction de la quantité de mémoire installée. Le ROS accepte un minimum de 512 K et l'installation de mémoire vive supplémentaire par bans contigus de 32 K jusqu'à la quantité maximum de 640 K. La mémoire additionnelle est mesurée selon la procédure suivante : l'adresse du segment de chacun des blocs de 32 K de RAM est écrite dans les deux premiers octets de chaque bloc respectif. Les segments sont alors vérifiés des adresses basses vers les adresses hautes jusqu'à la rencontre d'une adresse de segment ne correspondant pas, ou la rencontre du dernier bloc. Le registre Write Status 2 prend la valeur définie dans la table RAM0-RAM4 de la section 1.8.3.

8. Initialisation de la zone des variables du ROS dans la RAM système

Le ROS utilise les variables dans la zone 00300 à 00500. Reportez-vous à la section 2.4 pour une description complète de toutes ces variables et leurs valeurs d'initialisation.

9. Initialisation des 32 premiers vecteurs d'interruption

Les 32 premiers vecteurs d'interruption sont initialisés pour pointer vers les routines du ROS (voir tableau ci-dessous). Les routines d'interruptions logicielles qui n'exécutent aucune action pointent simplement vers une instruction IRET (Return from Interrupt) alors que les interruptions matérielles dans le même cas pointent vers une routine HWIRET qui envoie une fin d'interruption non spécifique au 8259 (contrôleur d'interruptions) et exécute une instruction IRET.

Interrup.	Objet	Type
0	Division par 0	Hard (HWIRET)
1	Pas à pas	Hard (HWIRET)
2	Routine d'erreur de parité (NMI)	Hard
3	Break	Hard (HWIRET)
4	Overflow	Hard (HWIRET)
5	Impression d'écran	Soft
6	Contrôle du bouton de la souris	Soft
7	Réservé	Soft (IRET)
8 (IRQ0)	Interruption horloge système	Hard
9 (IRQ1)	Interruption clavier	Hard
10 (IRQ2)	Interruption RTC	Hard (HWIRET)
11 (IRQ3)	COMMS	Hard (HWIRET)
12 (IRQ4)	COMMS	Hard (HWIRET)
13 (IRQ5)	Disque dur	Hard (HWIRET)
14 (IRQ6)	Routine d'interruption floppy	Hard
15 (IRQ7)	Interruption imprimante	Hard (HWIRET)
16	E/S Vidéo	Soft
17	Configuration système	Soft
18	Taille mémoire	Soft
19	E/S disques	Soft
20	E/S série	Soft
21	Fonctions avancées	Soft
22	E/S clavier	Soft

Interrup.	Objet	Type
23	E/S imprimante	Soft
24	Redémarrage système	Soft
25	Bootstrap disque	Soft
26	Horloge système et E/S RTC	Soft
27	Break clavier	Soft (IRET)
28	Interruption Ticker externe	Soft (IRET)
29	Table des paramètre vidéo	Vecteur soft
30	Table des paramètre disques	Vecteur soft
31	Table de matrice externe de caractères 8x8	Vecteur Soft

Les interfaces des vecteurs d'interruption ci-dessus sont décrites dans la section 2.3.

Le contrôleur d'interruptions programmable (PIC pour Programmable Interrupt Controller) 8259 est programmé pour que ses niveaux d'interruption IRQ0-IRQ7 utilisent les vecteurs d'interruption 8 à 15 du CPU comme nous l'indiquons dans la table ci-dessus.

10. Initialisation et test de l'interface disquettes

La fonction d'initialisation de l'interruption 19 est appelée, puis un contrôle des disquettes a lieu (voir 2.2.14).

11. Autotest du clavier

Le contrôleur du clavier retourne la valeur 0AAh après l'exécution avec succès de sa procédure d'autotest (voir 2.2.15).

12. Initialisation de l'affichage

Le test initial de la vidéo se fait à partir de la ROM IGA BIOS si elle est validée ou, sinon, le ROS effectue le contrôle comme décrit dans la section 1.8.2.

13. Initialisation du contrôleur d'interruptions 8259

Autorise les interruptions du 8259 aux niveaux 0 (compteur 0 du 8253), 1 (réception d'un code de scrutation du clavier) et 6 (contrôleur lecteur de disquette 765). Toutes les autres interruptions du 8259 sont masquées.

14. Affichage du message de signature du ROS

A la mise en route, le ROS fait un test de checksum de la NVR. Après l'affichage du message de copyright, le ROS affiche un avertissement si la somme de la NVR s'est révélée incorrecte. Si, en revanche, le test était positif (et si les dernières données d'initialisation étaient valides), l'ordinateur affiche la date et l'heure du dernier allumage de la machine.

15. Validation des NMI

Si une NMI advient, le traitement des interruptions par défaut du ROS provoque l'affichage d'un message d'erreur de parité de la RAM et plante le système. Cet état ne peut être interrompu que par la mise hors tension de la machine.

16. Initialisation de toutes les ROM externes

Le ROS recherche les ROM externes depuis l'adresse C0000 jusqu'à F4000 par sauts de 800h (2 K) et initialise les ROM répondant aux critères suivants :

1. Les deux premiers octets contiennent la valeur hexadécimale 55AA.
2. Les deux octets suivants indiquent la taille en multiples de 512 octets (1/2 K).
3. L'octet suivant est le point d'entrée de la routine d'initialisation.
4. L'octet de poids faible de la somme des octets de la ROM est zéro.

Lorsqu'une ROM répondant à ces spécifications est localisée, le ROS appelle son point d'entrée.

Si la somme de contrôle de la ROM externe n'est pas conforme, un message d'erreur est affiché et l'initialisation n'a pas lieu.

La ROM BIOS IGA respecte le standard que nous venons de décrire et si elle est validée, l'exécution de son point d'entrée (à C0000) est lancée. Elle effectue son autotest de mise en route (POST pour Power-On Self Test) et initialise :

256 K de mémoire vive vidéo.
Le contrôleur vidéo.
Elle charge les polices de caractères depuis des matrices de la ROM
Elle établit le mode d'affichage initial (par défaut) selon la configuration des interrupteurs.
Met en place l'interruption vidéo 16 selon le mode déterminé par les interrupteurs.

17. Vecteur d'initialisation sur disquette

Le logiciel en ROM tente de charger vers la mémoire (en 07C00h) le secteur d'initialisation de la disquette (bootstrap sector) depuis le lecteur A, face 0, piste 0, secteur 1. Si le vecteur de contrôle se charge correctement, il prend le contrôle grâce à un saut long (far jump) vers le segment 0000 offset 7C00. Sinon, après 10 tentatives, le ROS affiche un message demandant à l'utilisateur d'insérer une disquette système dans le lecteur A et d'appuyer sur une touche. Le ROS attend alors cet appui et répète la procédure de chargement.

2.2 AUTOTESTS DE MISE EN ROUTE

A l'allumage de la machine ou après un system reset, le ROS effectue une série de tests sur le matériel afin de vérifier son bon fonctionnement. Si un test échoue, le ROS affiche un message d'erreur sur le périphérique d'affichage principal et verrouille le système. L'interface clavier est traitée d'une autre manière en ce sens que le ROS répète l'autotest du clavier jusqu'à ce qu'il obtienne un résultat positif.

Le ROS effectue tous les tests à moins que les liaisons d'options (LK1-LK3) ne soient configurées en mode diagnostic (voir section 1.10.3), auquel cas seuls ont lieu les tests du clavier et des disquettes. En cas d'echec de l'un ou des deux contrôles, il y a affichage d'un message mais l'erreur est ignorée. Cela permet de poursuivre le diagnostic du système.

Après un reset logiciel (Control-Alt-Del), le ROS effectue tous les autotest à l'exception de ceux de la mémoire vive utilisateur et de la RAM vidéo. Le programme qui appelle l'initialisation du ROS après un reset logiciel doit écrire la valeur 1234h en 0:478h pour notifier au ROS qu'un reset logiciel a été demandé. Lorsque 1234h est reconnu, un reset hardware complet est exécuté ce qui réinitialise toutes les cartes des périphériques externes du connecteur d'extensions.

Avec la valeur spéciale 1235h, le ROS effectue les tests de mise en route comme décrit ci-dessus mais cela inhibe le reset hardware. Après cela, le ROS donne la valeur 1234h au mot situé en 0:488.

2.2.1 Procédure de test

En premier lieu, le ROS exécute les fonctions nécessaires à la mise en service de l'affichage soit de la carte IGA, soit d'un autre adaptateur vidéo principal ; il inscrit ensuite le message *"Patientez"* sur la première ligne de l'écran pour indiquer que le processus de test est en cours. A chaque début de test individuel (voir ci-dessous), il affiche un point.

Les tests ont lieu dans l'ordre suivant :

1. Checksum du ROS.
2. Test du contrôleur DMA (Direct Memory Access) 8237.
3. Test du timer 8253.
4. Test de l'interface périphérique programmable 8255.
5. Test de l'horloge temps réel HD 146818.
6. Test du 8250 pour communications asynchrones.
7. Test du port parallèle pour imprimante.
8. Test des registres de comptage X et Y de la souris.

9. Test de la RAM du système.
10. Test du contrôleur d'interruptions programmable 8259.
11. Test des lecteurs de disquette.
12. Test de l'interface clavier.

Le ROS utilise la pile durant les tests des disquettes, de l'interface clavier et du contrôleur d'interruptions programmable. Tous les autres tests se déroulent sans toucher ni à la pile ni à des variables en mémoire vive.

2.2.2 Méthodes de test

La plupart des tests de diagnostic des "périphériques" consistent en un test de la transmission des données et un test de décodage.

Test de transmission :

Il vérifie la fiabilité des opérations de lecture/écriture entre le CPU et un organe particulier. Une valeur test est écrite vers le périphérique et est relue puis comparée. Les valeurs modèles sont les suivantes :

Tous bits à 0
Tous bits à 1
Glissement d'un bit unique sur un octet, puis complément sur 8 bits.

Test de décodage :

Il détecte les erreurs de décodage d'adresses d'un organe matériel. Il consiste à choisir une adresse donnée d'un device, à écrire une valeur test (habituellement 0FFFFh) et à vérifier qu'elle puisse être relue. Ce test s'effectue à la fois en ordre ascendant et en ordre descendant (tous deux séquentiellement) afin de vérifier que la logique de décodage d'adresses soit correcte.

2.2.3 Checksum du ROS

Tous les octets de la ROM du système d'exploitation résident sont additionnés et l'on vérifie que l'octet de poids faible de la somme soit égal à 0. Si la vérification échoue, un message d'erreur indiquant une mauvaise somme de contrôle de la ROM est affiché.

2.2.4 Test du contrôleur DMA

Le test de décodage ascendant/descendant permet de s'assurer qu'on peut adresser le contrôleur DMA. S'il est négatif, un message indiquant une erreur DMA s'affiche.

2.2.5 Test du timer d'intervalle programmable

Le premier test auquel est soumis le 8253 est un test de transmission des données vers le compteur 2. On vérifie ensuite que la cadence de comptage du compteur 1 est correcte. Si l'un ou l'autre de ces tests échoue, un message d'erreur indiquant un dysfonctionnement du timer d'intervalle est affiché.

2.2.6 Test de l'interface de périphérique programmable

Le contrôle du 8255 consiste en un test de transmission de données vers les deux canaux d'état système (Status-1 et Status-2). La sortie 2 du timer 8253 est aussi vérifiée. Si l'un ou l'autre de ces tests échoue, un message d'erreur approprié est affiché.

2.2.7 Test de l'horloge temps réel

On vérifie que le deuxième compteur RTC (Real Time Clock) est correctement cadencé puis on effectue un test de transmission sur l'octet de checksum de la NVR (il est restauré). Si l'un ou l'autre de ces tests échoue, un message indiquant que l'horloge temps réel ne fonctionne pas correctement est affiché.

2.2.8 Test du circuit de communication série

Ce test confirme le bon fonctionnement en émission et réception du 8250 (c'est-à-dire du port série du système). Notez que ce test n'a pas une valeur absolue, il indique surtout que le 8250 fonctionne en mode diagnostic.

Le 8250 est configuré en boucle, à 9600 bauds, 8 bits de données, 1 bit de stop, pas de parité. Deux valeurs test sont envoyées et comparées aux valeurs reçues. Puis on interroge le registre d'état pour détecter une erreur de framing, de dépassement ou de parité. Si les valeurs test reçues sont différentes de celles qui ont été envoyées ou si une erreur apparaît au niveau du registre d'état, le message d'erreur du port série est affiché.

2.2.9 Test du port parallèle

Un test de transmission est exécuté sur la voie des données (data latch) de l'imprimante. Si une valeur test incorrecte est retournée, le message d'erreur de port parallèle est affiché.

2.2.10 Test des registres compteurs X et Y

Les registres X et Y sont mis à 0 puis relus pour vérifier qu'ils renferment bien cette valeur. Si le test échoue, le message de registre des coordonnées de la souris défectueux est affiché.

2.2.11 Test de la RAM du système

La quantité de mémoire de la zone utilisateur est mesurée selon la méthode décrite en 2.1 (on doit toujours obtenir 640 K). Le test de transmission de données est lancé vers toute la mémoire disponible suivi par un test de décodage ascendant/descendant. Si l'un ou l'autre échoue, il y a affichage du message d'erreur de RAM.

2.2.12 Test du contrôleur d'interruption programmable

Le test du 8259 consiste en un essai de transmission de données vers le registre d'interruption masquable et d'un test de reconnaissance d'interruption pour confirmer que celle-ci a bien eu lieu et peut être utilisée. En cas d'échec, le message de contrôleur d'interruption défectueux est affiché.

2.2.13 Test des lecteurs de disquettes

Il tente de vérifier que les lecteurs de disquettes installés se positionnent correctement. Le test déplace la tête de lecture/écriture à la piste 10 de chaque lecteur. Le ROS ne vérifie pas que la piste visée a réellement été atteinte. Si une erreur intervient, le message de contrôleur de lecteurs de disquettes défectueux s'affiche.

2.2.14 Test de l'interface clavier

Après une mise sous tension ou un reset, un autotest du clavier (effectué par le logiciel résident de celui-ci) a lieu. Le clavier retourne la valeur 0AAh si l'opération s'est déroulée de manière satisfaisante. Si une autre valeur que 0AAh est reçue, le message d'erreur de clavier est affiché et un reset du clavier est provoqué (ce qui relance la procédure d'autotest du clavier). Le cycle est répété jusqu'à ce que le test soit positif. Lorsque le test réussit, le message est effacé et l'ensemble du test est considéré comme bon. Durant les opérations d'autotest du clavier, un bip court est émis toutes les deux secondes pour en indiquer le déroulement.

2.3 LES INTERRUPTIONS DU ROS

Les 32 premiers vecteurs d'interruption sont mis en place par la procédure d'initialisation de mise en route. Les points d'entrée IRET logicielles et HWIRET matérielles sont des routines vides qui ne nécessitent ni condition d'entrée ni condition de sortie et ne sont pas détaillées ici.

Toute application qui aurait remplacé un vecteur d'interruption par son propre point d'entrée ne doit pas faire appel elle-même à une interruption quelconque du ROS à partir de sa propre routine d'interruption.

2.3.1 Interruption 2 : Erreur de parité (NMI)

La routine correspondant à une interruption 2 traite les erreurs de parité de la RAM du système. L'écran prend le mode d'affichage par défaut, il est effacé et un message d'erreur de parité de la RAM y est émis. La machine ne peut plus être utilisée tant qu'on ne l'éteint puis ne la rallume pas.

Cette routine n'utilise pas la RAM pour une pile ou des variables.

Une application se servant du coprocesseur 8087 doit fournir une routine d'interruption 2 pour le 8087.

Utilisation des registres du CPU :

Entrée : Pas de conditions.

Sortie : N'existe pas.

2.3.2 Interruption 5 : Impression d'écran (Print Screen)

La routine d'interruption 5 envoie le contenu de l'écran en mode caractères vers le port imprimante primaire. Etant donné qu'il s'agit d'une opération en mode caractères, l'impression d'un écran graphique risque de donner des résultats inattendus. Les caractères qui ne peuvent être relus à l'écran en mode graphique (par la fonction lecture de caractères de l'interruption vidéo 16) sont imprimés comme des espaces.

Si une impression d'écran est déjà en cours, l'interruption n'a pas d'effet.

La variable Print Screen Status (à 00500) est mise à 1 tant que l'impression est en cours. Lorsqu'elle se termine, la variable reprend la valeur 0. Si l'impression est abandonnée à cause d'un dépassement de temps de réponse (timeout), la variable est mise à 255.

Utilisation des registres du CPU :

Entrée : Pas de condition.

Sortie : Tous les flags et les registres sont préservés.

2.3.3 Interruption 6 : Contrôle des boutons souris

La routine d'interruption 6 du ROS fournit les routines utilitaires par défaut de la souris. Le logiciel résident du clavier génère un ensemble de codes de touches quand l'un ou les deux boutons de la souris est pressé ou relâché. Quand la routine d'interruption du clavier reconnaît un code de touche de la souris, elle appelle l'interruption 6. S'il s'agit d'un code d'établissement, la routine par défaut du ROS obtiendra le code clavier approprié par la NVR et retournera cette valeur avec le Carry flag mis. Si au contraire, il s'agit d'un code de relâchement, le bit de retenue sera effacé.

Utilisation des registres du CPU :

Entrée : Registre AL = code touche de la souris

Le bit 0 détermine le bouton :
0 = Bouton souris M1 (M pour mouse).
1 = Bouton souris M2.

Le bit 7 indique la pression ou le relâchement :
0 = pression (établissement)
1 = relâchement (rupture)

Sortie : Action spécifique du Carry et du registre AX

Carry SET : Insère un code clavier dans le tampon clavier

AX = code clavier à insérer

Carry CLEAR : Pas d'action à effectuer
AX est indéterminé

Tous les autres flags sont indéterminés.
Tous les autres registres sont préservés.

NOTE : Un code clavier de valeur FFFFh est ignoré et n'est pas recopié dans le tampon clavier.

2.3.4 Interruption 8 : Interruption de l'horloge système

La routine d'interruption 8 est appelée par l'horloge système (compteur 0 du 8253). Cette routine par défaut du ROS effectue les tâches suivantes :

1. Incrémentation du compteur sur 32 bits de l'horloge système. Ce compteur se trouve en RAM à l'adresse 0046C. Si l'horloge atteint 24 heures (01800B0h), le compteur est remis à zéro et le flag 24 heures (en 00470) est mis à 0FFh.

2. Si l'octet de poids faible du compteur de l'horloge système vaut 0, l'heure et la date courantes de l'horloge temps réel (RTC) sont copiées dans la NVR. La dernière heure recopiée à partir de la RTC avant que la machine soit mise hors tension sera affichée à la prochaine mise en route.

3. Si le compteur du temps de réaction des moteurs des lecteurs de disquette n'est pas égal à 0, il est décrémenté d'une unité. Si le compteur atteint 0, tous les moteurs des lecteurs sont arrêtés.

Utilisation des registres du CPU :

Entrée : Pas de condition.
Sortie : Tous les flags et registres sont préservés.

2.3.5 Interruption 9 : Interruption clavier

L'interruption hard du clavier est ainsi traitée par le ROS : lecture d'un code de touche depuis le clavier, traduction en un code clavier (token) sur 16 bits à l'aide d'une table interne et écriture dans le tampon clavier. Si le tampon est plein, alors le code clavier est rejeté et il y a émission d'un signal sonore par le haut-parleur. L'encodage des touches revient essentiellement à cela : l'octet haut est le numéro de la touche et l'octet bas est le code ASCII de la touche. Pour les touches n'ayant pas d'équivalent ASCII, le code clavier consiste en une valeur pour l'octet haut alors que l'octet bas est égal à zéro.

Entrée : Pas de condition.
Sortie : Tous les registres et flags sont préservés.

Code	Nom (GB) de la touche	Normal	ALT	CTRL	SHIFT	Num Lock
01	ESC	011B	Ignoré	011B	011B	N/A
02	1 et !	0231	7800	Ignoré	0221	N/A
03	2 et "	0332	7900	0300	0340	N/A
04	3 et £	0433	7A00	Ignoré	0423	N/A
05	4 et $	0534	7B00	Ignoré	0524	N/A
06	5 et %	0635	7C00	Ignoré	0625	N/A
07	6 et ^	0736	7D00	071E	075E	N/A
08	7 et &	0837	7E00	Ignoré	0826	N/A
09	8 et *	0938	7F00	Ignoré	092A	N/A
0A	9 et (	0A39	8000	Ignoré	0A28	N/A
0B	0 et)	0B30	8100	Ignoré	0B29	N/A
0C	- et _	0C2D	8200	0C1F	0C5F	N/A
0D	= et +	0D3D	8300	Ignoré	0D2B	N/A
0E	<-DEL	0E08	Ignoré	0E7F	0E08	N/A
0F	TAB	0F09	Ignoré	Ignoré	0F00	N/A
10	Q	1071	1000	1011	1051	N/A

11	W	1177	1100	1117	1157	N/A
12	E	1265	1200	1205	1245	N/A
13	R	1372	1300	1312	1352	N/A
14	T	1474	1400	1414	1454	N/A
15	Y	1579	1500	1519	1559	N/A
16	U	1675	1600	1615	1655	N/A
17	I	1769	1700	1709	1749	N/A
18	O	186F	1800	180F	184F	N/A
19	P	1970	1900	1910	1950	N/A
1A	[et {	1A5B	Ignoré	1A1B	1A7B	N/A
1B	] et }	1B5D	Ignoré	1B1D	1B7D	N/A
1C	CR ↵	1C0D	Ignoré	1C0D	1C0D	N/A
1D	CTRL	Ignoré	Ignoré	—	Ignoré	N/A
1E	A	1E61	1E00	1E01	1E41	N/A
1F	S	1F73	1F00	1F13	1F53	N/A
20	D	2064	2000	2004	2044	N/A
21	F	2166	2100	2106	2146	N/A
22	G	2267	2200	2207	2247	N/A
23	H	2368	2300	2308	2348	N/A
24	J	246A	2400	240A	244A	N/A
25	K	256B	2500	250B	254B	N/A
26	L	266C	2600	260C	264C	N/A
27	; et :	273B	Ignoré	Ignoré	273A	N/A
28	' et @	2827	Ignoré	Ignoré	2822	N/A
29	# et ~	2960	Ignoré	Ignoré	297E	N/A
2A	SHIFT Gauche	Ignoré	Ignoré	Ignoré	—	N/A
2B	\ et |	2B5C	Ignoré	2B1C	2B7C	N/A
2C	Z	2C7A	2C00	2C1A	2C5A	N/A
2D	X	2D78	2D00	2D18	2D58	N/A
2E	C	2E63	2E00	2E03	2E43	N/A
2F	V	2F76	2F00	2F16	2F56	N/A
30	B	3062	3000	3002	3042	N/A
31	N	316E	3100	310E	314E	N/A
32	M	326D	3200	320D	324D	N/A
33	, et <	332C	Ignoré	Ignoré	333C	N/A
34	. et >	342E	Ignoré	Ignoré	343E	N/A
35	/ et ?	352F	Ignoré	Ignoré	353F	N/A
36	SHIFT Droite	Ignoré	Ignoré	Ignoré	—	N/A
* 37	* et PRTSC	372A	Ignoré	7200	Prnt Scrn	N/A
38	ALT	Ignoré	—	Ignoré	Ignoré	N/A
39	ESPACE	3920	3920	3920	3920	N/A
* 3A	CAPS LOCK	Ignoré	Ignoré	Ignoré	Ignoré	N/A

3B	F1	3B00	6800	5E00	5400	N/A
3C	F2	3C00	6900	5F00	5500	N/A
3D	F3	3D00	6A00	6000	5600	N/A
3E	F4	3E00	6B00	6100	5700	N/A
3F	F5	3F00	6C00	6200	5800	N/A
40	F6	4000	6D00	6300	5900	N/A
41	F7	4100	6E00	6400	5A00	N/A
42	F8	4200	6F00	6500	5B00	N/A
43	F9	4300	7000	6600	5C00	N/A
44	F10	4400	7100	6700	5D00	N/A
* 45	NUM LOCK	Ignoré	Ignoré	PAUSE	Ignoré	N/A
* 46	SCRL LOCK	Ignoré	Ignoré	BREAK	Ignoré	N/A
47	KEY PAD 7	4700	Ignoré	7700	N/A	4737
48	KEY PAD 8	4800	Ignoré	Ignoré	N/A	4838
49	KEY PAD 9	4900	Ignoré	8400	N/A	4939
4A	KEY PAD -	4A2D	Ignoré	Ignoré	N/A	4A2D
4B	KEY PAD 4	4B00	Ignoré	7300	N/A	4B34
4C	KEY PAD 5	Ignoré	Ignoré	Ignoré	N/A	4C35
4D	KEY PAD 6	4D00	Ignoré	7400	N/A	4D36
4E	KEY PAD +	4E2B	Ignoré	Ignoré	N/A	4E2B
4F	KEY PAD 1	4F00	Ignoré	7500	N/A	4F31
50	KEY PAD 2	5000	Ignoré	Ignoré	N/A	5032
51	KEY PAD 3	5100	Ignoré	7600	N/A	5133
52	KEY PAD 0	5200	Ignoré	Ignoré	N/A	5230
53	KEY PAD .	5300	Ignoré	Ignoré	N/A	532E
54-6F	NON DEFINI	Ignoré	Ignoré	Ignoré	Ignoré	Ignoré
* 70	DEL ->	N/A	N/A	N/A	N/A	N/A
71-73	NON DEFINI	Ignoré	Ignoré	Ignoré	Ignoré	Ignoré
* 74	ENTER	N/A	N/A	N/A	N/A	N/A
75-76	NON DEFINI	Ignoré	Ignoré	Ignoré	Ignoré	Ignoré
* 77	JOY FIRE2	N/A	N/A	N/A	N/A	N/A
* 78	JOY FIRE1	N/A	N/A	N/A	N/A	N/A
* 79	JOY DROITE	4D00	4D00	4D00	4D00	4D00
* 7A	JOY GAUCHE	4B00	4B00	4B00	4B00	4B00
* 7B	JOY BAS	5000	5000	5000	5000	5000
* 7C	JOY HAUT	4800	4800	4800	4800	4800
* 7D	SOURIS M2	N/A	N/A	N/A	N/A	N/A
* 7E	SOURIS M1	N/A	N/A	N/A	N/A	N/A
7F	NON DEFINI	Ignoré	Ignoré	Ignoré	Ignoré	Ignoré

Les contacts du joystick produisent les codes curseur respectifs.

Les codes de touches marqués d'un '*' entraînent un traitement spécial décrit ci-dessous.

Physiquement, les entrées de la table où figure le mot "ignored" se distinguent à cause du bit de poids fort mis à 1. Le contrôleur du clavier ignore ces combinaisons de touches.

2.3.5.1 Touches à traitement spécial

Quelques touches et quelques combinaisons de touches font l'objet d'un traitement spécial que nous décrivons ci-dessous. Sauf précision contraire, elles n'entraînent pas l'écriture d'un code clavier dans le tampon.

1. [Ctrl]+[Alt]+[Del] : Reset

Lorsqu'un reset est détecté, cela entraîne l'exécution d'un reset soft du système. Le processus d'initialisation de mise en route est reconduit mais sans les tests de la RAM système et de la RAM vidéo.

2. [Ctrl]+[Num Lock] : Pause

Le ROS attend l'appui sur une autre touche (sauf [Ctrl]+[Num Lock]). L'exécution de l'application est suspendue. Etant donné que le firmware du clavier allume le voyant de la touche [Num Lock], le ROS bascule le flag Num Lock pour que le clavier et l'unité centrale soient en phase.

3. [Ctrl]+[Scroll Lock] : Break

Lorsque le Break est détecté, l'interruption 27 est exécutée et le tampon clavier est effacé. Le code clavier 0000h est inséré dans le tampon.

4. [Shift]+[PrtSc] : Print Screen (Impression de l'écran)

Quand Print Screen est détecté, l'interruption 5 (impression de l'écran du ROS) est appelée. Si ce point d'entrée est maintenu, l'impression d'écran continuera à marcher. Certains systèmes d'exploitation comme CP/M et Dos Plus installent un pointeur nul à cet endroit.

5. [Ins] : Bascule du mode insertion

Chaque fois que le code de la touche [Ins] (52) est reçu, sauf en mode Num Lock, le bit du mode insertion (bit 7 en 00417 de la RAM) est inversé.

6. [Scroll Lock] : Bascule du mode scroll

Chaque fois que l'on appuie sur la touche [Scroll Lock], le bit du mode Scroll (bit 4 en 00417 de la RAM) est inversé. Notez que la combinaison [Ctrl]+[Scroll Lock] (qui correspond à Break) n'affecte pas le mode Scroll.

7. [Caps Lock] : Bascule mode majuscules

Chaque fois que la touche [Caps Lock] est pressée, le bit de bascule du mode majuscules (bit 6 de l'adresse 00417) est inversé.

8. [Num Lock] : Bascule mode numérique

Chaque fois que l'on appuie sur la touche [Num Lock], le bit Num Lock (bit 5 de l'adresse 00417) est inversé.

9. [Alt]+[Touche du pavé numérique de 0 à 9] : Code caractère absolu

En maintenant la touche [Alt] enfoncée, on peut entrer un code clavier absolu en le tapant sur les touches numériques du pavé numérique. L'appui sur toute autre touche ramène le code clavier à une valeur nulle (et insère dans le tampon le code clavier de la touche "alternée").

Au moment où la touche [Alt] est relâchée, le code clavier modulo 256 est placé dans le tampon. Un code 0 est ignoré.

10. [Enter] et boutons Feu

Lorsque l'on appuie sur [Enter] (code 74) et l'un ou les deux boutons Feu du joystick (78 ou 79), le code clavier est obtenu par l'intermédiaire de la table de correspondance de la NVR et ajouté au tampon clavier.

11. Boutons de la souris

Le logiciel résident du clavier génère quatre codes de touches pour indiquer l'appui ou le relâchement de l'un des boutons de la souris. Lorsque le ROS reçoit l'un de ces codes en provenance du clavier, il fait un appel long à l'adresse présente dans le vecteur d'interruption des boutons de la souris (interruption 6). A l'initialisation de mise en route ou de reset, une routine ROS par défaut est chargée pour ce vecteur. Cette routine force le ROS à insérer dans le tampon clavier un code clavier qu'il va chercher dans la NVR. Ce code dépend évidemment du bouton qui a été pressé (M1 ou M2). Les codes touches correspondant à un relâchement du bouton sont ignorés.

12. [Del ->] : Forward Delete

Quand le code touche d'effacement avant (code 70) est reçu, le code clavier correspondant est obtenu via la NVR et est placé dans le tampon.

13. [Alt], [Ctrl], [Shift], [Caps Lock] et [Num Lock]

La correspondance entre les différents codes de touches et leurs codes claviers associés se fait en tenant compte de l'état des touches [Alt], [Ctrl], [Shift], [Caps Lock] et [Num Lock]. Cet état est représenté en 00417. Lorsque l'on appuie sur la touche [Shift], cela inverse l'état des flags Caps Lock et Num Lock. Si plusieurs de ces touches sont maintenues pressées en même temps, leur ordre de priorité est le suivant : Alt, puis Ctrl, puis Caps Lock ou Num Lock.

Lorsque Caps Lock est actif, il convertit les codes de touches alphabétiques (a-z) en leur équivalent majuscules.

Note :

Certains systèmes d'exploitation (comme Dos Plus) installent leurs propres points d'entrée dans les vecteurs d'interruption et ces routines peuvent donner des résultats différents de ceux décrits ici.

2.3.6 Interruption 14 : Contrôleur des lecteurs de disquettes

La routine utilitaire du ROS correspondant à l'interruption 14 met le bit 7 du RAM DRIVE RESTORE FLAG à 1, ce qui indique qu'une interruption du contrôleur de lecteurs de disquettes est advenue.

Utilisation des registres du CPU :

Entrée : Pas de condition.

Sortie : Tous les flags et les registres sont préservés

2.3.7 Interruption 16 : E/S vidéo compatibles 6845 (int 10h)

La routine utilitaire ROS de l'interruption 16 fournit un ensemble de routines pour la lecture et l'écriture de caractères en modes texte et graphique. En mode graphique, les caractères sont dessinés à partir d'une matrice (voir section 2.3.21). L'interruption 16 traite aussi le scrolling haut et bas, la lecture et l'affichage de pixels (mode graphique CGA uniquement) et la lecture du crayon optique.

Il est important de noter que ce point d'entrée de l'interruption n'est actif qu'en modes compatibles 6845 ou avec une carte graphique CGA, MDA ou Hercules installée. Il n'est pas fait appel à la ROM BIOS IGA. Les appels compatibles avec le mode d'émulation EGA sont détaillés en section 2.3.8. Il faut remarquer d'autre

part que pour Int 16 les registres d'appel sont pratiquement les mêmes pour la gestion des modes vidéo et les affichages de chaînes et de caractères. Cela tient au fait que le mode compatible EGA, qui a été implémenté postérieurement, présente une compatibilité logicielle avec la mise en place des registres d'appel du mode graphique concerné par cette interruption-ci. On entre dans les modes d'affichage compatibles avec le 6845 en mettant ON le sw5 ou grâce à l'utilitaire DISPLAY. A l'appel de ces modes, le vecteur de l'interruption vidéo 16 (à 0:40h) est modifié pour pointer sur le point d'entrée fixe (à F000:F065) de l'interruption vidéo du ROS, ce qui active ces routines.

Une configuration supplémentaire de l'IGA doit avoir eu lieu pour accepter la compatibilité 6845.

Utilisation des registres du CPU en mode compatible 6845 :

Entrée : AH = numéro de fonction :

0 - Détermine le mode vidéo
1 - Fixe la taille du curseur
2 - Fixe la position du curseur
3 - Donne la position du curseur
4 - Donne la position du crayon optique
5 - Détermine la page d'écran
6 - Scrolling vers le haut
7 - Scrolling vers le bas
8 - Lecture de caractère et attributs
9 - Ecriture de caractère et attributs
10 - Lecture de caractère seulement
11 - Chargement du Color Select Register
12 - Ecriture d'un pixel
13 - Lecture d'un pixel
14 - Ecriture d'un caractère en mode émulation Télétype
15 - Donne les paramètres vidéo

Tous les autres registres prennent les valeurs requises par la fonction.

Sortie : Si le numéro de fonction est supérieur à 15, alors l'indicateur de retenue (carry) est mis, sinon il est effacé.

Tous les autres flags et registres prennent une valeur déterminée par la fonction.

Les modes texte 0 et 1 requièrent 2000 octets de RAM vidéo alors que les modes texte 2 et 3 en demandent 4000. Le ROS profite de la place disponible en mode texte pour fournir plusieurs pages écran. Cela signifie que l'application peut préparer un certain nombre de pages d'affichage et passer de l'une à l'autre en fonction de ses besoins.

En général, les paramètres passés aux routines du ROS ne sont pas vérifiés et l'on doit prendre garde à l'envoi de valeurs non prévues qui peuvent engendrer des résultats erratiques. En particulier, méfiez-vous des valeurs limites. Par exemple, rendre égaux le haut et le bas de la fenêtre d'affichage pour les fonctions 6 et 7 de l'interruption 16 crée en fait un affichage d'une seule ligne mais cela posera des problèmes au moment de l'exécution d'un scrolling.

Fonction 0 de l'interruption vidéo 16 du ROS :
Détermine le mode vidéo (Set Video Mode)

Utilisation des registres du CPU :

Entrée : AH = 0

AL = Sélection du mode :

0 - Texte 25 lignes, 40 colonnes
1 - Idem mode 0
2 - Texte 25 lignes, 80 colonnes
3 - Idem mode 2
4 - Graphique 320 pixels x 200 lignes, palette 1
5 - Graphique 320 pixels x 200 lignes, palette 2
6 - Graphique 640 pixels x 200 lignes, 2 couleurs
7 - Texte 25 lignes, 80 colonnes avec carte Monochrome

Sortie : Tous les flags et les registres sont préservés.

En mode 4, on peut choisir la palette 0 en modifiant le registre de sélection de couleur avec la fonction 11 de Int 16. La section 1.11.4 Graphiques Basse résolution (320 x 200) traite des définitions de palettes.

Le passage en mode 5 doit être suivi de la sélection de la palette zéro (voir Registre de sélection de couleur en 1.11.10.2) pour valider la palette 2. Si le mode d'affichage par défaut est celui d'une carte monochrome externe, le mode 7 est sélectionné quel que soit le mode spécifié par AL.

Voici comment le ROS effectue la sélection du mode vidéo :

1. Il inhibe la sortie vidéo.

2. Il attribue les valeurs ligne 0, Colonne 0 aux curseurs de toutes les pages.

3. Il écrit le mode dans le registre de sélection de mode (3B8/3D8).

4. Il recharge les registres CRTC 6845 depuis la table des paramètres vidéo (Video Parameter Table) grâce au vecteur d'interruption 31.

5. Il efface la RAM vidéo soit avec des espaces (code ASCII 020h) et l'octet d'attribut par défaut conservé par la NVR pour le mode texte, soit avec des zéros en mode graphique.

6. S'il y a un affichage couleur, il charge le registre Color Select (3D9h) : il donne au bord d'écran la couleur de fond par défaut. En mode graphique, il attribue la surbrillance aux couleurs d'avant-plan. En mode 6 (mode graphique 640) l'avant-plan reçoit la couleur blanche.

7. Pour les modes textes, il sélectionne la page 0.

8. Il définit la taille du curseur avec le départ du curseur en ligne 6 et l'arrivée du curseur en ligne 7.

9. Il autorise la sortie vidéo.

Fonction 1 de l'interruption vidéo 16 du ROS :
Fixe la taille du curseur

Cette fonction ne s'applique qu'aux modes texte, les modes graphiques n'admettant pas de curseur. Elle attribue les valeurs de départ et d'arrivée du curseur (cursor start et cursor end) au CRTC.

Utilisation des registres du CPU :

Entrée : AH = 1
CH = Ligne (en pixels) de départ du curseur (de 0 à 31)
CL = Ligne (en pixels) d'arrivée du curseur (de 0 à 31)

Sortie : Tous les flags et les registres sont préservés.

Si l'on donne la valeur de départ 31, le ROS l'interprète comme un ordre d'extinction du curseur. De plus, la fonction d'inhibition hardware du curseur du 6845 est autorisée lorsque le bit 5 du registre de départ du curseur est à 1.

Fonction 2 de l'interruption vidéo 16 du ROS :
Fixe la position du curseur

Cette fonction permet de placer le curseur à la ligne et à la colonne voulues de la page spécifiée.

Utilisation des registres du CPU :

Entrée : AH = 2
BH = Numéro de page pour modes 0 à 3
(doit être à 0 pour les autres modes)
DH = Ligne où placer le curseur
DL = Colonne où placer le curseur

Sortie : Tous les flags et les registres sont préservés.

La table de la section 1.11.7 donne les valeurs possibles des numéros de pages, de colonnes et de lignes.

Fonction 3 de l'interruption vidéo 16 du ROS :
Donne la position du curseur

Cette fonction retourne les numéros de ligne et de colonne où se trouve le curseur à l'intérieur de la page indiquée.

Utilisation des registres du CPU :

Entrée : AH = 3
BH = Numéro de page pour modes 0 à 3
(doit être à 0 pour les autres modes)

Sortie : DH = Numéro de ligne
DL = Numéro de colonne
CH = Ligne (de balayage) de départ du curseur
CL = Ligne (de balayage) d'arrivée du curseur

Tous les autres registres et flags sont préservés.

Reportez-vous à la section 1.11.7 pour obtenir les numéros de pages valides et les limites de position de curseur.

Fonction 4 de l'interruption vidéo 16 du ROS :
Donne la position du crayon optique

Cette fonction retourne la position du crayon optique.

Utilisation des registres du CPU :

Entrée : AH = 4

Sortie : Si l'utilisateur appuie sur le bouton du crayon, alors :
AH = 1
DH = Numéro de ligne (en caractères) de 0 à 24
DL = Numéro de colonne (en caractères) de 0 à 79

CH = Numéro de ligne (en pixels) de 0 à 199
BX = Numéro de colonne (en pixels) de 0 à 639

Si l'on ne presse pas le bouton du crayon :
AH = 0
BX, CH et DX préservés

Dans les deux cas :
Tous les autres flags et registres sont préservés.

Fonction 5 de l'interruption vidéo 16 du ROS : Fixe la page écran

Cette fonction active l'une des pages écran virtuelles.

Utilisation des registres du CPU :

Entrée : AH = 5
AL = Numéro de la page à afficher

Sortie : Tous les flags et les registres sont préservés.

Reportez-vous à la section 1.11.7 pour obtenir les numéros de pages valides et leurs limites de position de curseur.

Fonction 6 de l'interruption vidéo 16 du ROS : Scrolling haut

Cette fonction fait défiler vers le haut et du nombre de lignes indiqué tout ou partie de l'affichage de la page écran active.

Utilisation des registres du CPU :

Entrée : AH = 6
DH = Numéro de ligne du bas de la zone à scroller
DL = Dernière colonne (à droite) de la zone à scroller
CH = Numéro de ligne du haut de la zone à scroller
CL = Première colonne (à gauche) de la zone à scroller
BH = Attribut des lignes vierges insérées sous la zone scrollée
AL = Décalage en nombre de lignes
Si AL = 0, efface la zone spécifiée.

Si AL est différent de 0, la zone spécifiée est décalée du nombre de lignes figurant dans AL.
Si DH = CH alors AL doit être égal à zéro.

Sortie : Tous les registres sont préservés.
Carry est effacé et tous les autres flags sont indéterminés.

Le scrolling agit toujours sur la page d'écran active.

Le scrolling hardware n'est pas possible. Un scrolling s'effectue en copiant des zones de la RAM vidéo.

Pour afficher la couleur du fond, les lignes vierges sont remplies de zéros en modes graphiques.

Notez que cette routine ne fonctionnera pas correctement si, à l'entrée, CH = DH avec AL différent de 0. Cela est vrai pour tous les environnements de ROM compatibles.

Fonction 7 de l'interruption vidéo 16 du ROS : Scrolling bas

Cette fonction fait défiler vers le bas et du nombre de lignes indiqué tout ou partie de l'affichage de la page écran active.

Utilisation des registres du CPU :

Entrée : AH = 7
DH = Numéro de ligne du bas de la zone à scroller
DL = Dernière colonne (à droite) de la zone à scroller
CH = Numéro de ligne du haut de la zone à scroller
CL = Première colonne (à gauche) de la zone à scroller
BH = Attribut des lignes vierges insérées au dessus de la zone scrollée
AL = Décalage en nombre de lignes
Si AL = 0, efface la zone spécifiée.
Si AL est différent de 0, la zone spécifiée est décalée du nombre de lignes figurant dans AL.
Si DH = CH alors AL doit être égal à zéro.

Sortie : Tous les flags et les registres sont préservés.

Le scrolling agit toujours sur la page écran active.

Le scrolling hardware n'est pas assuré. Le scrolling s'effectue par copie de RAM d'affichage.

Notez que cette routine ne fonctionne pas correctement si l'on donne à l'entrée CH = DH avec AL différent de zéro.

Fonction 8 de l'interruption vidéo 16 du ROS :
Lit un caractère et ses attributs

Cette fonction lit le caractère sous le curseur de la page spécifiée et indique son octet d'attribut.

Dans les modes graphiques, les caractères sont générés à partir d'une matrice interne si leur code est compris entre 0 et 127 ou à partir d'une table externe si le code est compris entre 128 et 255. L'adresse de cette dernière matrice figure dans le vecteur d'interruption 31. Reportez-vous à 2.3.23 pour plus de détails.

Utilisation des registres du CPU :

Entrée : AH = 8
BH = Page pour les modes texte 0 à 3
(doit être à 0 pour les autres modes)

Sortie : AL = Caractère
AH = Octet d'attribut. (Inchangé en mode graphique)

Tous les autres flags et registres sont préservés.

Fonction 9 de l'interruption vidéo 16 du ROS :
Ecriture d'un caractère et de ses attributs

Cette fonction écrit un caractère (ou un ensemble de caractères identiques) et ses attributs à la position courante du curseur de la page spécifiée.

Dans les modes graphiques, les caractères sont générés à partir d'une matrice interne si leur code est compris entre 0 et 127 ou à partir d'une table externe si le code est compris entre 128 et 255. L'adresse de cette dernière matrice figure dans le vecteur d'interruption 31. Reportez-vous à 2.3.23 pour plus de détails.

Utilisation des registres du CPU :

Entrée : AH = 9
AL = Caractère à écrire
BH = Page où écrire pour les modes texte 0 à 3
(doit être à 0 pour les autres modes)
BL = En modes texte
Attributs du caractère
En modes graphiques
Le mode d'écriture est ainsi défini :
Bit 7 = 0 : remplacement
Bit 7 = 1 : mode caractère XOR
Bits 0 et 1 : couleur caractère
(pour modes 4 et 5 seulement)
CX = Nombre de répétitions du caractère

Sortie : Tous les flags et les registres sont préservés.

Le nombre de répétitions définit le nombre d'adresses consécutives où il faut écrire le caractère et l'attribut. Dans les modes graphiques, tous les caractères doivent tenir sur la ligne courante.

Dans les modes graphiques, si le bit 7 de BL est à 1 alors l'écriture se fait en mode OU exclusif avec l'affichage précédent de cet emplacement.

Fonction 10 de l'interruption vidéo 16 du ROS :
Ecriture d'un caractère seulement

Cette fonction écrit un caractère (ou plusieurs fois le même caractère) à la position courante du curseur de la page désignée. En modes texte, les attributs courants de l'emplacement ne seront pas modifiés.

Dans les modes graphiques, les caractères sont générés à partir d'une matrice interne si leur code est compris entre 0 et 127 ou à partir d'une table externe si le code est compris entre 128 et 255. L'adresse de cette dernière matrice figure dans le vecteur d'interruption 31. Reportez-vous à 2.3.23 pour plus de détails.

Utilisation des registres du CPU :

Entrée : AH = 10
AL = Caractère à écrire
BH = Page où écrire pour les modes texte 0 à 3
(doit être à 0 pour les autres modes)
BL = En modes texte :
non utilisé
En modes graphiques
Le mode d'écriture est ainsi défini :
Bit 7 = 0 : remplacement
Bit 7 = 1 : mode XOR
Bits 0 et 1 : couleur caractère
(pour modes 4 et 5 seulement)
CX = Nombre de répétitions du caractère

Sortie : Tous les flags et les registres sont préservés.

Le nombre de répétitions définit le nombre d'adresses consécutives où il faut écrire le caractère.

Dans les modes graphiques, si le bit 7 de BL est à 1 alors l'écriture se fait en mode OU exclusif avec l'affichage précédent de cet emplacement.

Fonction 11 de l'interruption vidéo 16 du ROS :
Chargement du Color Select Register

Cette fonction écrit les bits IRGB du Color Select Register compatible CGA ou les bits de sélection de la palette.

Utilisation des registres du CPU :

Entrée : AH = 11
BH = Fonction :
0 - Configure les bits IRGB (3-0) comme spécifié par BL.
Autre valeur - Détermine la palette (0 ou 1) en fonction de la valeur de BL

Sortie : Tous les flags et les registres sont préservés.

La modification de la palette (BH différent de 0) n'a d'effet qu'en modes 4 et 5 (modes 320 pixels). Reportez-vous à la section 1.11.8 pour les aspects hardware.

Fonction 12 de l'interruption vidéo 16 du ROS : Ecriture d'un pixel

Cette fonction écrit un pixel donné (ne fonctionne qu'en mode graphique).

Utilisation des registres du CPU :

Entrée : AH = 12
DX = Ligne du pixel (0 à 199)
CX = Colonne du pixel (0 à 639)
AL = Mode d'écriture :
Bit 7 : 0 remplacement
Bit 7 : 1 mode XOR
Bits 0 et 1 : Couleur de caractère pour les modes graphiques 320

Sortie : Tous les flags et les registres sont préservés.

La couleur de caractère donnée en AL doit être comprise entre 0 et 3 en modes 4 et 5 (modes 320 pixels) et doit être 0 ou 1 pour le mode 6 (mode 640 pixels).

Fonction 16 de l'interruption vidéo 16 du ROS : Lecture d'un pixel

Cette fonction sert à lire la valeur d'un pixel individuel (dans les modes graphiques seulement).

Utilisation des registres du CPU :

Entrée : AH = 13
DX = Ligne du pixel (0 à 199)
CX = Colonne du pixel (0 à 639)

Sortie : AL = Couleur du caractère
Tous les autres flags et registres sont préservés.

Fonction 16 de l'interruption vidéo 16 du ROS :
Ecriture en mode émulation TTY

Cette fonction écrit un caractère en mode émulation Télétype à l'emplacement du curseur de la page active.

Utilisation des registres du CPU :

Entrée : AH = 14
AL = Caractère à écrire
BL = En modes texte,
BL n'est pas utilisé
En modes graphiques :
Bit 7 = 0 : remplacement
Bit 7 = 1 : mode XOR
Bits 0 et 1 = Couleur du caractère
(en modes 4 et 5 uniquement)

Sortie : Tous les flags et les registres sont préservés.

Après écriture, le numéro de colonne du curseur est augmenté de 1. Si l'emplacement du curseur vient à dépasser la fin de la ligne, le numéro de colonne est remis à 0 et le numéro de ligne incrémenté. Si le numéro de ligne incrémenté dépasse le numéro de la dernière ligne visible, alors la page entière est scrollée d'une ligne.

Dans les modes texte, la ligne vierge apportée au bas de la page est constituée d'espaces portant le même attribut que le premier caractère de la ligne précédente. En mode graphique, elle est faite de 0.

L'écriture des caractères spéciaux suivants entraîne un traitement spécial :

BEL (07h) Emission d'un son court par le haut-parleur
BS (08h) Décrémente d'une colonne la position du curseur à moins qu'il ne soit déjà à la colonne 0, auquel cas la fonction est ignorée.
CR (0Dh) Met le curseur à la colonne 0.
LF (0Ah) Incrémente d'une ligne la position du curseur et déclenche éventuellement la procédure de scrolling décrite ci-dessus.

Tous les autres caractères de contrôle sont affichés.

Fonction 15 de l'interruption vidéo 16 du ROS :
Donne les paramètres vidéo courants

Cette fonction retourne le mode vidéo courant, la page active, ainsi que le nombre de colonnes visibles.

Utilisation des registres du CPU :

Entrée : AH = 15

Sortie : BH = Page active (ou 0 si l'on se trouve en mode graphique ou en mode texte 7).
AH = Nombre de colonnes visibles (40 ou 80).
AL = Mode vidéo courant (0 à 7).

Tous les autres flags et registres sont préservés.

2.3.8 Interruption 16 du BIOS IGA :

E/S vidéo compatibles EGA (int 10h)

Cette routine propose un vaste ensemble d'utilitaires qui sont par beaucoup d'aspects similaires à celles de la ROM ROS mais qui apportent des fonctions supplémentaires tirant profit de l'environnement hardware amélioré.

Il faut noter que ce point d'entrée d'interruption est actif avec les modes d'affichage compatibles EGA et que certains paramètres peuvent différer des appels standard. Il est donc important de se reporter à la bonne section lorsque l'on fait référence à l'interruption 16. Les appels concernant les modes compatibles CGA, MDA ou Hercules sont traités en 2.3.7. Remarquons que les registres à l'appel de Int 16 sont pratiquement les mêmes pour les appels de modes vidéo et d'affichage de caractères. Lorsque l'on entre dans les modes d'affichage compatibles EGA soit par le positionnement de sw5 (OFF), soit grâce à l'utilitaire DISPLAY, le vecteur de l'Int 16 vidéo est modifié pour pointer vers l'interruption vidéo du BIOS IGA dont les routines deviennent actives. D'autres configurations hardware d'IGA doivent avoir eu lieu pour assurer la compatibilité EGA.

Utilisation des registres du CPU en mode compatible EGA :

Entrée : AH = Numéro de fonction comme ci-dessous

0 - Fixe le mode vidéo
1 - Détermine la taille du curseur
2 - Positionne le curseur
3 - Donne la position du curseur
4 - Donne la position du crayon optique
5 - Détermine la page active
6 - Scrolling haut
7 - Scrolling bas
8 - Lecture de caractère et d'attribut
9 - Ecriture de caractère et d'attribut
10 - Ecriture de caractère seulement
11 - Fixe la palette

12 - Ecriture d'un pixel
13 - Lecture d'un pixel
14 - Ecriture d'un caractère en mode émulation Télétype
15 - Donne les paramètres vidéo
16 - Attribue une valeur au Palette Register
17 - Charge le Character Generator
18 - Retourne EGC State et fixe le vecteur PrtSc
19 - Ecriture d'une chaîne

Tous les autres registres prennent les valeurs requises par la fonction.

Sortie : Si le numéro de fonction est supérieur à 19, alors l'appel est rejeté et n'a pas d'effet.

Tous les flags sont préservés et les registres prennent une valeur déterminée par la fonction.

En général, les paramètres passés à une fonction ne sont pas vérifiés et il faut être prudent, certaines valeurs fantaisistes risquant de produire des résultats inattendus.

Fonction 0 de l'interruption vidéo 16 de l'IGA :
Détermine le mode vidéo (Set Video Mode)

Entrée : AH = 0
AL = Sélection du mode :

0 - Texte 25 lignes par 40 colonnes N/B
1 - Texte 25 lignes par 40 colonnes couleurs
2 - Texte 28 lignes par 80 colonnes N/B
3 - Texte 28 lignes par 80 colonnes couleurs
4 - Graphique 320 pixels x 200 lignes avec palette 1
5 - Graphique 320 pixels x 200 lignes avec palette 2
6 - Graphique 640 pixels x 200 lignes, 2 couleurs
7 - Texte 25 lignes par 80 colonnes en Monochrome
13 - Graphique 320 pixels x 200 lignes, 16 couleurs
14 - Graphique 640 pixels x 200 lignes, 16 couleurs
15 - Graphique 640 pixels x 350 lignes, Monochrome
16 - Graphique 640 pixels x 350 lignes, 16/64 couleurs

Sortie : Tous les flags et les registres sont préservés.

La grande différence entre les désignations N/B et Couleurs des modes 0-3 ne tient pas aux couleurs affichables mais au fait que les modes couleurs disposent d'un "flicker wait" pour éviter un effet de neige avec certains périphériques. En réalité, l'Int 16 du BIOS IGA peut contrôler une carte CGA à condition qu'elle soit installée comme carte secondaire (configuration des switches, section 1.22).

Si le mode par défaut est celui d'une carte monochrome, le mode 7 est sélectionné sans tenir compte de la valeur de AL à moins que celle-ci soit 15d auquel cas le mode monochrome graphique 640x350 est activé.

Les modes 0 à 7 sont classés dans les modes compatibles, alors que les modes 13 à 16 sont des modes étendus. Les modes 8 à 12 ne sont pas accessibles à l'utilisateur, certains d'entre eux servant au chargement en RAM de polices de caractères.

Fonction 1 de l'interruption vidéo 16 de l'IGA :
Fixe la taille du curseur

Cette fonction n'est destinée qu'aux modes texte, les modes graphiques n'acceptant pas de curseur. Elle permet de fixer les valeurs de départ et d'arrivée du curseur CRTC.

Entrée : AH = 1
CH = Ligne (de balayage) de départ du curseur
(de 0 à 31)
CL = Ligne (de balayage) d'arrivée du curseur
(de 0 à 31)

Sortie : Tous les flags et les registres sont préservés.

Des valeurs supérieures à 32 et inférieures ou égales à 64 éteignent le curseur. Donner une valeur de début de curseur supérieure à celle de fin de curseur fait généralement disparaître celui-ci ou peut lui faire prendre un aspect étrange.

Fonction 2 de l'interruption vidéo 16 de l'IGA :
Fixe la position du curseur

Cette fonction permet de placer le curseur à la ligne et à la colonne voulues dans la page spécifiée.

Utilisation des registres du CPU :

Entrée : AH = 2
BH = Numéro de page
DH = Ligne où placer le curseur
DL = Colonne où placer le curseur

Sortie : Tous les flags et les registres sont préservés.

La table de la section 1.11.7 donne les valeurs possibles des numéros de pages, de colonnes et de lignes.

Fonction 3 de l'interruption vidéo 16 de l'IGA :
Donne la position du curseur

Cette fonction retourne les numéros de ligne et de colonne où se trouve le curseur à l'intérieur de la page indiquée.

Utilisation des registres du CPU :

Entrée : AH = 3
BH = Numéro de page

Sortie : DH = Numéro de ligne
DL = Numéro de colonne
CH = Ligne (en pixels) de départ du curseur
CL = Ligne (en pixels) d'arrivée du curseur

Tous les autres registres et flags sont préservés.

La table de la section 1.11.7 donne le détail de ces valeurs et les limites de positionnement du curseur.

Fonction 4 de l'interruption vidéo 16 de l'IGA :
Donne la position du crayon optique

Cette fonction retourne la position du crayon optique.

Utilisation des registres du CPU :

Entrée : AH = 4

Sortie : Si l'utilisateur appuie sur le bouton du crayon, alors :
AH = 1
DH = Numéro de ligne (en caractères) de 0 à 24
DL = Numéro de colonne (en caractères) de 0 à 79
CH = Numéro de ligne (en pixels) de 0 à 199 pour modes 0-7
CL = Indéterminé si mode = 0 à 7
CX = Numéro de ligne (en pixels) de 0 à 349 pour les modes 13-16
BX = Numéro de colonne (en pixels) de 0 à 639
Si l'on ne presse pas le bouton du crayon :
AH = 0
BX, CH et DX préservés
Dans les deux cas :
Tous les autres flags et registres sont préservés.

Fonction 5 de l'interruption vidéo 16 de l'IGA : Fixe la page écran

Cette fonction détermine la page d'écran active.

Utilisation des registres du CPU :

Entrée : AH = 5
AL = Numéro de la page à afficher

Sortie : Tous les flags et les registres sont préservés.

Les pages sont numérotées de 0 à n-1 où n est le nombre de pages disponibles. Reportez-vous à la section 1.11.7 pour obtenir les numéros de pages valides et leurs limites de position de curseur.

Fonction 6 de l'interruption vidéo 16 de l'IGA : Scrolling haut

Cette fonction fait défiler vers le haut et du nombre de lignes indiqué tout ou partie de l'affichage de la page écran active.

Utilisation des registres du CPU :

Entrée : AH = 6
DH = Numéro de ligne du bas de la zone à scroller
DL = Dernière colonne (à droite) de la zone à scroller
CH = Numéro de ligne du haut de la zone à scroller
CL = Première colonne (à gauche) de la zone à scroller
BH = Attribut des lignes vierges insérées sous la zone scrollée

AL = Décalage en nombre de lignes
Si AL = 0, efface la zone spécifiée.
Si AL est différent de 0, la zone spécifiée est décalée vers le haut du nombre de lignes figurant dans AL.
Si DH = CH alors AL doit être égal à zéro.

Sortie : Tous les flags et les autres registres sont préservés.

Le scrolling agit toujours sur la page d'écran active.

Le scrolling hardware n'est pas possible. Un scrolling s'effectue en copiant des zones de la RAM vidéo.

Pour afficher la couleur du fond, les lignes vierges sont remplies de zéros en modes graphiques.

Fonction 7 de l'interruption vidéo 16 de l'IGA : Scrolling bas

Cette fonction fait défiler vers le bas et du nombre de lignes indiqué tout ou partie de l'affichage de la page écran active.

Utilisation des registres du CPU :

Entrée : AH = 7
DH = Numéro de ligne du bas de la zone à scroller
DL = Dernière colonne (à droite) de la zone à scroller
CH = Numéro de ligne du haut de la zone à scroller
CL = Première colonne (à gauche) de la zone à scroller
BH = Attribut des lignes vierges insérées au dessus de la zone scrollée
AL = Décalage en nombre de lignes
Si AL = 0, efface la zone spécifiée.
Si AL est différent de 0, la zone spécifiée est décalée du nombre de lignes figurant dans AL.
Si DH = CH alors AL doit être égal à zéro.

Sortie : Tous les flags et les autres registres sont préservés.

Le scrolling agit toujours sur la page écran active.

Le scrolling hardware n'est pas assuré. Le scrolling s'effectue par copie de RAM d'affichage.

Fonction 8 de l'interruption vidéo 16 de l'IGA :
Lit un caractère et ses attributs

Cette fonction lit le caractère sous le curseur de la page spécifiée et indique son octet d'attribut.

Dans les modes graphiques, les caractères sont générés à partir d'une matrice interne si leur code est compris entre 0 et 127 ou à partir d'une table externe si le code est compris entre 128 et 255. L'adresse de cette dernière matrice figure dans le vecteur d'interruption 31. Reportez-vous à 2.3.23 pour plus de détails.

Utilisation des registres du CPU :

Entrée : AH = 8
BH = Page où effectuer la lecture

Sortie : AL = Caractère
AH = Octet d'attribut. (Inchangé en mode graphique)

Tous les flags et les autres registres sont préservés.

Reportez-vous aux sections 1.11.1 et 1.11.2 pour la définition des octets d'attributs.

Reportez-vous à la section 1.11.7 pour obtenir les numéros de pages valides.

Fonction 9 de l'interruption vidéo 16 de l'IGA :

Cette fonction écrit un caractère (ou un ensemble de caractères identiques) et ses attributs à la position courante du curseur de la page spécifiée.

Dans les modes compatibles CGA (3 à 5), les caractères sont générés à partir de matrices. On trouve un vecteur d'adresse sur 20 bits dans le vecteur d'interruption 67 pour les caractères de 0 à 127. Pour les autres (128 à 255), l'adresse de la table se trouve dans le vecteur d'interruption 31. La section 2.3.23 explique cette organisation.

Pour les modes compatibles EGA étendus (13-16), le vecteur 67 (en 10Ch) pointe vers une matrice où figurent les 256 caractères. Ce vecteur est initialisé par la ROM BIOS IGA durant l'initialisation de mise en route et pointe vers une table de caractères en ROM (de C2000 à C7FFF).

Utilisation des registres du CPU :

Entrée : AH = 9
AL = Caractère à écrire
CX = Nombre de répétitions
BH = Page où écrire
BL = En modes texte
Attributs du caractère

En modes graphiques
Le mode d'écriture est ainsi défini
Bit 7 = 0 : remplacement
Bit 7 = 1 : mode caractère XOR
Bits 0 à 3 : couleur caractère

Sortie : Tous les flags et les registres sont préservés.

Le nombre de répétitions définit le nombre d'adresses consécutives où il faut écrire le caractère et l'attribut. Dans les modes graphiques, tous les caractères doivent tenir sur la ligne courante.

Dans les modes graphiques, si le bit 7 de BL est à 1 alors l'écriture se fait en mode OU exclusif avec l'affichage précédent de cet emplacement.

La couleur de caractère donnée en BL doit se trouver dans la fourchette 0 à 3 pour les modes graphiques 320x200, 0 à 15 pour les modes 640 pixels avec 16 couleurs, ou doit être égale à 0 ou 1 pour le mode N/B en 640 de large.

Fonction 10 de l'interruption vidéo 16 de l'IGA :
Ecriture d'un caractère seulement

Cette fonction écrit un caractère (ou plusieurs fois le même caractère) à la position courante du curseur de la page désignée. En modes texte, les octets d'attribut de tous les caractères écrits seront inchangés.

Reportez-vous à la fonction 9 ci-dessus pour l'organisation des matrices de caractères.

Utilisation des registres du CPU :

Entrée : AH = 9
AL = Caractère à écrire
CX = Nombre de répétitions
BH = Page où écrire
BL = En modes texte
BL n'est pas utilisé

En modes graphiques
Le mode d'écriture est ainsi défini
Bit 7 = 0 : remplacement
Bit 7 = 1 : mode XOR
Bits 0 à 3 : couleur caractère

Sortie : Tous les flags et les registres sont préservés.

Le nombre de répétitions définit le nombre d'adresses consécutives où il faut écrire le caractère et l'attribut. Dans les modes graphiques, tous les caractères doivent tenir sur la ligne courante.

Dans les modes graphiques, si le bit 7 de BL est à 1 alors l'écriture se fait en mode OU exclusif (XOR) avec l'affichage précédent de cet emplacement.

La couleur de caractère donnée en BL doit se trouver dans la fourchette 0 à 3 pour les modes graphiques 320x200, 0 à 15 pour les modes 640 pixels avec 16 couleurs, ou doit être égale à 0 ou 1 pour le mode N/B en 640 de large.

Fonction 11 de l'interruption vidéo 16 de l'IGA :
Détermine la palette

Cette fonction attribue des valeurs à la palette de couleurs compatible CGA.

Entrée : AH = 11
BH = Numéro de fonction :
0 - Donne au fond la couleur déterminée par BL
Autre valeur - Détermine la palette (0 ou 1) en fonction de la valeur de BL

Sortie : Tous les flags et les registres sont préservés.

Le changement de numéro de palette (BH différent de 0) ne s'applique qu'aux modes graphiques 320x200. Reportez-vous à la section 1.11.2 pour les informations sur le hardware.

Fonction 12 de l'interruption vidéo 16 de l'IGA :
Ecriture d'un pixel

Cette fonction écrit un pixel individuel (n'est valide qu'en mode graphique).

Utilisation des registres du CPU :

Entrée : AH = 12
DX = Ligne du pixel
CX = Colonne du pixel
AL = Mode d'écriture et pixel
Bit 7 : 0 remplacement
Bit 7 : 1 mode XOR
Bits 0 à 3 = Donnée pixel

Sortie : Tous les flags et les registres sont préservés.

La valeur de pixel donnée en AL doit se trouver dans la fourchette 0 à 3 pour les modes graphiques 320x200, 0 à 15 pour les modes 640 pixels avec 16 couleurs, ou doit être égale à 0 ou 1 pour le mode N/B en 640 de large.

Fonction 13 de l'interruption vidéo 16 de l'IGA :
Lecture d'un pixel

Cette fonction sert à lire la valeur d'un pixel individuel en modes graphiques.

Entrée : AH = 13
DX = Ligne du pixel
CX = Colonne du pixel

Sortie : AL = Couleur du pixel
Tous les autres registres sont préservés.
Tous les flags sont préservés.

La couleur retournée obéira aux mêmes règles que celles valant pour l'écriture d'un pixel (fonction 12 ci-dessus).

Fonction 14 de l'interruption vidéo 16 de l'IGA :
Ecriture en mode émulation TTY

Cette fonction écrit un caractère en mode émulation télétype à l'emplacement du curseur de la page active.

Utilisation des registres du CPU :

Entrée : AH = 14
AL = Caractère à écrire
BL = En modes texte,
BL n'est pas utilisé
En modes graphiques :
Bit 7 = 0 : remplacement
Bit 7 = 1 : mode XOR
Bits 0 3 = Couleur du caractère

Sortie : Tous les flags et les registres sont préservés.

Les bits de couleur du caractère sont les mêmes que pour la fonction 10 (écriture d'un caractère) décrite plus haut.

Après écriture, le numéro de colonne du curseur est augmenté de 1. Si l'emplacement du curseur vient à dépasser la fin de la ligne, le numéro de colonne est remis à 0 et le numéro de ligne incrémenté. Si le numéro de ligne incrémenté dépasse le numéro de la dernière ligne visible, alors la page entière est scrollée d'une ligne. Dans les modes texte, la ligne vierge apportée au bas de la page est constituée d'espaces portant le même attribut que le premier caractère de la ligne précédente. En mode graphique, elle est faite de 0.

L'écriture des caractères spéciaux suivants entraîne un traitement spécial :

BEL (07h) Emission d'un son court par le haut-parleur
BS (08h) Décrémente d'une colonne la position du curseur à moins qu'il ne soit déjà à la colonne 0, auquel cas la fonction est ignorée.
CR (0Dh) Met le curseur à la colonne 0.

LF (0Ah) Incrémente d'une ligne la position du curseur et déclenche éventuellement la procédure de scrolling décrite ci-dessus.

Tous les autres caractères de contrôle sont affichés.

Fonction 15 de l'interruption vidéo 16 de l'IGA :
Donne les paramètres vidéo courants

Cette fonction retourne le mode vidéo courant, la page active, ainsi que le nombre de colonnes visibles.

Utilisation des registres du CPU :

Entrée : AH = 15

Sortie : BH = Numéro de page active
AH = Nombre de colonnes visibles (40 ou 80)
AL = Mode vidéo courant
Tous les autres registres sont préservés.
Tous les flags sont préservés.

Fonction 16 de l'interruption vidéo 16 de l'IGA :
Set Palette Register(s)

Cette fonction permet d'affecter les valeurs voulues aux registres individuels de la palette, au registre Hi-Res Border (overscan), à tous les registres à la fois ou de manipuler le bit d'intensité/clignotement.

Utilisation des registres du CPU :

Entrée : AH = 16
AL = Numéro de sous-fonction :

0 - Détermine un registre individuel
1 - Affecte une valeur au registre Hi-Res Border
2 - Fixe tous les registres et Hi-Res Border
3 - Modifie le bit intensité/clignotement

Sortie : Tous les flags et les registres sont préservés.

Pour la sous-fonction 0 :
BL = Numéro du registre de la palette (0-15)
BH = Valeur à attribuer (0-63)

Pour la sous-fonction 1 :
BH = Valeur pour le registre Hi-Res Border

Pour la sous-fonction 2 :
ES:DX = Pointeur vers une table de 17 octets composée ainsi :
0 à 15 = Valeurs des registres de la palette (de 0 à 15)
16 = Valeur pour le registre Hi-Res Border

Pour la sous-fonction 3 :
BL = 1 - Autorise le clignotement (Inhibe la double intensité de l'avant-plan).
0 - Inhibe le clignotement (Rétablit la double intensité de l'avant-plan).

Fonction 17 de l'interruption vidéo 16 de l'IGA :
Chargement d'une matrice de caractères

Cette fonction permet la mise en place d'un jeu de caractères étendu.

Utilisation des registres du CPU :

Entrée : AH = 17
AL = Sous-fonction comme suit :

0 - Charge la matrice utilisateur
1 - Charge la fonte en ROM 8x14
2 - Charge la fonte en ROM 8x8
3 - Modifie le registre Character Set Select (registre de sélection du jeu de caractères)
16 - Charge la matrice utilisateur
17 - Charge la fonte en ROM 8x14
18 - Charge la fonte en ROM 8x8
32 - Etablit la matrice vidéo 8x8 (Int Vect 31)

33 - Etablit le vecteur de la matrice utilisateur Hi-Res étendue
34 - Etablit le vecteur de la matrice en ROM Hi-Res étendue 8x8
35 - Etablit le vecteur de la matrice en ROM Hi-Res étendue 8x14
48 - Retourne le vecteur du jeu de caractères (Character Set Vector)

Sortie : Tous les flags et les registres sont préservés (sauf pour la sous-fonction 48).

Les sous-fonctions 0 à 2 réalisent la même chose que les sous-fonctions 16 à 18 à la différence que les premières provoquent un changement de mode complet (sans effacement de la RAM vidéo) alors que les sous-fonctions 16 à 18 reprogramment les paramètres du curseur CRTC sur la base de la taille des caractères (Points). Ces options chargent une matrice de caractères dans la zone mémoire qui est réservée à cet effet sur les plans 2 et 3.

Reportez-vous à la description du Character Set Select Register du chapitre 1 pour l'organisation des matrices de caractères étendues et leur sélection par le bit d'attribut 3.

Pour la sous-fonction 0 :

AL = 0 (Charge la matrice utilisateur)
BL = Fonte à charger (0-3)
DX = Offset de départ à l'intérieur de la fonte (0-255 caractères)
BH = Points (Octets par caractère) (1-31)
CX = Nombre de caractères à charger (1-256)
ES:BP = Pointeur sur l'adresse de la matrice utilisateur (User Font Table)

Pour la sous-fonction 1 :

AL = 1 (Charge la matrice ROM 8x14)
BL = Fonte à charger (0-3)

Pour la sous-fonction 2 :
AL = 2 (Charge la matrice ROM 8x8)
BL = Fonte à charger (0-3)

Pour la sous-fonction 3 :
AL = 3 (Registre Character Set Select)
BL = Sélection A/B sur 4 bits

Pour la manipulation de ce registre Character Set Select, consultez la section 1.11.8.

Pour la sous-fonction 16 :
AL = 16 (Charge la matrice utilisateur)

Pour les autres registres, consultez les explications relatives à la sous-fonction 0. Comme nous l'indiquions plus haut, cette sous-fonction-ci ne diffère de la n°0 que parce que le mode courant est maintenu mais avec un curseur du CRTC dont la valeur dépend de la hauteur (en ligne de balayage) du caractère sélectionné. Cette sous-fonction et les deux suivantes doivent être appelées tout de suite après un changement de mode et la reprogrammation du CRTC qui en résulte doit être la plus proche possible de la configuration originale sinon vous pourriez obtenir des effets indésirables.

Pour la sous-fonction 17
AL = 17 (Charge la matrice ROM 8x8)
BL = Fonte à charger (0-3)

Reportez-vous à la section 1.11.8 pour les valeurs à donner au Character Set Select Register.

Pour la sous-fonction 32 :
AL = 32 (Matrice Video 8x8)
ES:BP = Pointeur vers l'adresse de la User Graphics Font pour les caractères 128 à 255. Voir 2.3.22 Matrice Vidéo

Pour la sous-fonction 33 :
AL = 33 (User Hi-Res Graphics Characters Vector)
ES:BP = Pointeur vers l'adresse de la User Graphics Font pour les caractères de 0 à 255.

CX = Nombre d'octets par caractère (1-31)

BL = 0 - Le nombre de lignes (ROWS) est spécifié par DL
1 - 14 lignes
2 - 25 lignes
3 - 43 lignes

Cette matrice sert à former les caractères de l'écran lorsque l'on se trouve sous l'un des modes graphiques étendus haute résolution.

La valeur courante de Rows (-1) se trouve à l'adresse système 0:484 et le nombre d'octets par caractère (Points) en 0:485. Le pointeur du vecteur Int 67 est en 0:10C.

Pour la sous-fonction 48 :
AL = 48 (Return Character Set Vector)
BH = 0 - Retourne le vecteur 31 (128 caractères supérieurs de la matrice utilisateur 8x8)
1 - Retourne le vecteur 67 (256 caractères de la matrice Extended Hi-Res)
2 - Retourne l'adresse de la fonte ROM 8x14
3 - Retourne l'adresse de la fonte ROM 8x8
4 - Retourne l'adresse d'origine des 128 caractères supérieurs de la fonte 8x8 en ROM
5 - Retourne l'adresse de la fonte 9x14 en ROM

Au retour de la sous-fonction 48 :
CX = Nombre d'octets par caractère (Points en 0:485)
DL = ROWS - 1 (tiré de 0:484)
ES:BP = Vecteur demandé

Sortie : Tous les flags et les autres registres sont préservés.

La table 9x14 est une table combinée : le premier octet représente le numéro de caractère et est suivi de 14 octets formant la matrice du caractère. La table se termine par un 0.

Fonction 18 de l'interruption vidéo 16 de l'IGA :
Retourne EGC Status et fixe le vecteur PrtSc

Cette fonction a deux actions possibles : soit elle retourne le registre EGC Status, soit elle sélectionne l'adresse de la routine d'impression d'écran du BIOS IGA en remplacement du vecteur PrtSc (5) d'origine.

Utilisation des registres du CPU :

Entrée : AH = 18
BL = 16 : retourne EGC Status
32 - Donne au vecteur PrtSc la valeur du point d'entrée de la routine d'impression d'écran de la ROM IGA BIOS

Sortie : Si l'on a demandé EGC Status :

BH = 0 : Mode couleur
1 : Mode Monochrome
BL = Nombre de blocs de 64 K de RAM vidéo moins 1 (donc '3' en général puisque le PC 1640 possède 256 K de RAM vidéo)
CL = Etat des switches 1 à 4 (depuis 0:488 inversé)
CH = Features (OFh : le connecteur Features n'est pas installé sur le PC 1640)

Tous les autres flags et registres sont préservés.

S'il s'agissait de modifier le vecteur PrtSc, tous les flags et les registres seraient préservés.

Fonction 19 de l'interruption vidéo 16 de l'IGA :
Ecriture d'une chaîne

Cette fonction écrit une chaîne dans la RAM vidéo depuis l'emplacement du curseur et dans la page spécifiée. La chaîne peut être constituée des caractères seuls (auquel cas l'octet d'attribut est fourni par BL) ou de couples caractère et attribut. Les caractères de contrôle entraînent le traitement spécial décrit dans la section

consacrée à la fonction 14 de l'interruption vidéo 16 de l'IGA. Ils ne sont pas associés à un octet d'attribut figurant dans la chaîne. La coupure de fin de ligne se fait en fonction de la largeur d'affichage indiquée par la variable système ROWS (0:488).

Utilisation des registres du CPU :

Entrée : AH = 19
AL = Fonction comme ci-dessous :

0 - La chaîne se compose de caractères uniquement et l'attribut est dans BL. Le curseur restera à la position précédant l'écriture de la chaîne.

1 - La chaîne se compose de caractères uniquement et l'attribut est dans BL. La position du curseur sera mise à jour après l'écriture.

2 - La chaîne se compose d'un (ou de plusieurs) couple caractère et attribut. Le curseur restera à la position précédant l'écriture de la chaîne.

3 - La chaîne se compose d'un (ou de plusieurs) couple(s) caractère et attribut. La position du curseur sera mise à jour après l'écriture.

BH = Page où s'effectuera l'écriture
DH = Ligne du curseur
DL = Colonne du curseur
CX = Nombre de caractères (non compris les octets d'attributs s'ils figurent dans la chaîne)
ES:BP = Pointeur vers la chaîne

Sortie : Tous les flags et les registres sont préservés.

2.3.9 Interruption 17 : Configuration système

Cette interruption soft retourne l'état courant de la configuration système tel qu'il est défini aux adresses RAM 00410 et 00411h (voir section 2.4).

Utilisation des registres du CPU :

Entrée : Pas de condition

Sortie : AX = Etat de la configuration

Bits	Fonction
14 & 15	Nombre d'imprimantes (1-3)
13	Inutilisé
12	Si = 1, une interface manette de jeux optionnelle est montée
11	Toujours 0
09 & 10	Nombre d'interfaces série (1 ou 2)
8	Inutilisé
7	Toujours 0
6	Si = 1, un second lecteur de disquette est monté
4 & 5	Mode d'affichage par défaut (DDM)
2 & 3	Toujours 1
1	Si = 1, un coprocesseur 8087 est installé
0	Toujours 1

Tous les flags et les autres registres sont préservés.

Vous trouverez les états des modes par défaut des bits DDM1 et DDM2 dans la section 1.8.2 (Port A - Status-1 Input).

2.3.10 Interruption 18 : Taille mémoire

Cette interruption soft retourne la taille mémoire (vive) du système telle qu'elle figure en 00413 et 00414h.

Utilisation des registres du CPU :

Entrée : Pas de condition.

Sortie : AX = Nombre de blocs de 1 K de RAM installée (640 pour un PC 1640)

Tous les flags et les autres registres sont préservés.

2.3.11 Interruption 19 : E/S lecteurs de disquette

Cette interruption soft assure les fonctions de lecture, écriture, vérification et formatage des lecteurs de disquette reliés au contrôleur standard.

En pratique, une fois que MS DOS a été chargé et qu'une application a été activée, le processus de startup du DOS aura sauvé le vecteur d'interruption du ROS (qui est en 0:4Ch) et installé son propre vecteur d'appel. DOS fait cela pour pouvoir faire face à des conditions particulières comme un DMA dépassant la limite de 64 K où il découpera cette requête d'E/S en plusieurs opérations. De plus, tout disque dur installé aura détourné vers lui le vecteur de cette interruption 19 de sorte que les requêtes d'E/S dont le bit de poids fort du numéro de drive sera à 1 seront traitées par les routines issues de la ROM du disque dur (généralement autour de C8000h). La plupart du temps, les paramètres d'appel des opérations sur disque dur sont les mêmes que ceux décrits ici pour la lecture et l'écriture mais les ROM BIOS du disque dur apportent des fonctionnalités supplémentaires. Les codes d'erreurs des disques durs sont parfois étendus à ceux des lecteurs de disquettes décrits ici. La liste des fonctions et des codes d'erreurs d'un disque dur compatible typique sont documentés après la description des fonctions ROS.

Utilisation des registres du CPU :

Entrée : AH = Fonction d'E/S disque souple :

0 - Initialise le sous-système disque.
1 - Retourne l'état de la dernière opération
2 - Lit un nombre de secteurs consécutifs
3 - Ecrit un nombre de secteurs consécutifs

4 - Vérifie un nombre de secteurs consécutifs
5 - Formate une piste

Sortie : AH = Octet de statut

0 - L'opération s'est déroulée avec succès
1 - Numéro de fonction (ou de drive) incorrect
2 - Marque d'adresse (Address Mark) manquante
3 - Disquette protégée (Ecriture ou Format seulement)
4 - Enregistrement non trouvé
8 - Overrun en DMA
9 - Tentative de DMA sur un segment de plus de 64 K
16 - Erreur CRC
32 - Erreur du contrôleur de disques souples
64 - Erreur de positionnement
128 - Timeout du contrôleur de disquettes (Drive Not Ready)

Tous les autres registres seront déterminés par la fonction.

Pour toutes les fonctions disques le Carry sera mis à 1 en cas d'erreur (et AH = numéro d'erreur), sinon il sera effacé. Tous les autres flags sont indéterminés.

Fonction disquettes 0 : Initialisation du sous-système disque

Cette fonction effectue une interruption totale de l'interface disque.

1. Reset du FDC (Floppy Disk Controller)
2. Reconfiguration des paramètres FDC en fonction de ceux figurant dans la table des paramètres disques (voir interruption 30)

Utilisation des registres du CPU :

Entrée : AH = 0

Sortie : AH/flags = Comme indiqué ci-dessous
Autres registres préservés.

Lorsqu'une erreur est retournée par toute autre fonction d'E/S des disquettes, la réinitialisation du sous-système doit être lancée avant l'appel d'une autre fonction disques.

Fonction disquettes 1 : Retourne le dernier état

Cette fonction retourne l'octet d'état et le bit Carry de la dernière opération d'E/S disquettes.

Entrée : AH = 1

Sortie : AH/flags = Etat de la dernière opération comme décrit précédemment.

Tous les autres registres sont préservés.

Fonction disquettes 2 : Lecture de secteurs

Cette fonction lit le nombre demandé de secteurs consécutifs. Tous les secteurs lus doivent être sur la même piste.

Utilisation des registres du CPU :

Entrée : AH = 2
DH = Numéro de tête (0 ou 1)
DL = Numéro de drive (0 ou 1)
CH = Numéro de piste (à partir de 0)
CL = Numéro de secteur de départ (à partir de 1)
BX = Adresse Offset du tampon de lecture
ES = Adresse Segment du tampon de lecture
AL = Nombre de secteurs à lire

Sortie : AH/flags : comme décrit précédemment
AL = Nombre de secteurs lus avec succès
(Indéterminé si erreur de Timeout)

Tous les autres registres sont préservés.

Fonction disquettes 3 : Ecriture de secteurs

Cette fonction écrit le nombre de secteurs consécutifs voulu. Ils doivent tous être sur la même piste.

Utilisation des registres du CPU :

Entrée : AH = 3
DH = Numéro de tête (0 ou 1)
DL = Numéro de drive (0 ou 1)
CH = Numéro de piste
CL = Numéro de secteur de départ
BX = Adresse Offset du tampon d'écriture
ES = Adresse Segment du tampon d'écriture
AL = Nombre de secteurs à écrire

Sortie : AH/flags : comme décrit précédemment
AL = Nombre de secteurs écrits avec succès
(Indéterminé si erreur de Timeout)
Tous les autres registres sont préservés.

Fonction disquettes 4 : Vérification de secteurs

Cette fonction vérifie le nombre de secteurs voulu. Ils doivent tous être sur la même piste.

Utilisation des registres du CPU :

Entrée : AH = 4
DH = Numéro de tête (0 ou 1)
DL = Numéro de drive (0 ou 1)
CH = Numéro de piste
CL = Numéro de secteur de départ
AL = Nombre de secteurs à vérifier

Sortie : AH/flags : comme décrit précédemment
AL = Nombre de secteurs vérifiés avec succès
(Indéterminé si erreur de Timeout)

Tous les autres registres sont préservés.

Etant donné que le processus de vérification est interrompu à la première erreur, AL représente le nombre de secteurs vérifiés avec succès avant l'occurrence de l'erreur ou le nombre de secteurs demandés s'ils étaient tous corrects.

Fonction disquettes 5 : Formatage d'une piste

Cette fonction formate une piste entière.

Utilisation des registres du CPU :

Entrée : AH = 5
DH = Numéro de tête (0 ou 1)
DL = Numéro de drive (0 ou 1)
CH = Numéro de piste
BX = Adresse Offset du tampon de formatage
ES = Adresse Segment du tampon de formatage

Sortie : AH/flags : comme décrit précédemment

Tous les autres registres sont préservés.

Le tampon de formatage comprend 4 octets d'information pour chaque secteur de la piste :

1. Numéro de piste.
2. Face.
3. Numéro de secteur.
4. Code de la taille du secteur :
   0 : 128 octets/secteur
   1 : 256 octets/secteur
   2 : 512 octets/secteur
   3 : 1024 octets/secteur

La longueur du GAP, l'octet de remplissage et le nombre de secteurs par piste nécessaires à la commande Format du contrôleur sont obtenus dans la table des paramètres disques (DPT). Voir section 2.3.22.

2.3.11.1 Paramètres d'appel et registres pour disque dur

Comme nous l'expliquions au début de la section consacrée à Int 19 pour les disquettes, la ROM d'un contrôleur de disque dur pour slot d'extension fournit une interface logicielle comparable à celle du contrôleur de disquettes mais avec des fonctions étendues nécessaires par l'adaptation à l'environnement disque dur. Cette section donne les paramètres d'appel pour un disque dur "compatible". Il faut absolument se méfier des appels de bas niveau avec les disques à technologies nouvelles (encodage, compression, taille variable des clusters, etc..), et donc en particulier ceux qui offrent des capacité de plus de 32 Mo.

Utilisation des registres du CPU :

Entrée : AH = Sélecteur de fonction disque dur

0 - Reset contrôleur
1 - Retourne l'état de la dernière opération
2 - Lit un nombre spécifié de secteurs consécutifs
3 - Ecrit un nombre spécifié de secteurs consécutifs
4 - Vérifie un nombre spécifié de secteurs consécutifs
5 - Formate une piste
6 - Formate une piste avec marquage de mauvais secteurs
7 - Formate tout le disque
8 - Retourne les paramètres du disque
9 - Etablit les paramètres du disque
10 - Lecture longue
11 - Ecriture longue
12 - Positionnement sur une piste
13 - Reset contrôleur (autre entrée)
14 - Lecture du tampon secteur
15 - Ecriture du tampon secteur
16 - Test de disque prêt
17 - Recalibrage du positionnement
18 - Diagnostic RAM
19 - Diagnostic disque
20 - Diagnostic contrôleur
22 - Parque les têtes

Sortie : AH = Octet d'état (Status byte)

0 - Opération achevée avec succès
1 - Fonction (ou drive) incorrecte
2 - Adress Mark manquante
4 - Enregistrement non trouvé
9 - Tentative de DMA sur un segment de plus de 64 K
11 - Marque de piste défectueuse détectée
16 - Erreur ECC
17 - Donnée ECC corrigée
32 - Erreur du contrôleur de disque dur
64 - Erreur de positionnement
128 - Timeout du contrôleur de disque (Drive Not Ready)
187 - Erreur indéfinie
255 - Echec de la lecture du Status

Les registres pour l'appel des fonctions disque dur ont des attributions très similaires à celles de l'appel des fonctions disquettes. Leurs configurations typiques sont les suivantes :

Numéro de fonction	AH:0-20 et 22 (selon le tableau précédent)
Nombre de secteurs	AL : 1-n (en général 17 secteurs/piste)
Facteur d'interleave	AL:1-16 (En général 3 à 7). Opérations de formatage.
Numéro de drive	DL : 80h-FFh (C: a le n°80h, D: a 81h, etc)
Numéro de secteur	CL : 1-17 (6 bits de poids faibles de CL)
Numéro de tête	DH : 0-n (souvent 3 pour les 2O Mo à 4 têtes)
Numéro de cylindre	CH : 0-1023 (8 bits de poids faible et 2 bits supplémentaires de poids fort dans les 2 bits les plus forts de CL)
Adresse tampon	ES:BX : Segment:Offset

Sortie : AH : Retourne le code d'erreur comme indiqué plus haut
CF : Carry clear - Pas d'erreur, AH = 0
Carry set - Code d'erreur dans AH

Lorsque l'on demande les paramètres disque (fonction 8), les registres de retour sont les suivants :

Numéro de disque dur	DL (1 à n)
Numéro de tête maximum	DH (0 à n-1)
Secteurs par piste	CL (bits 0 à 5)
Numéro de cylindre maximum	CH (10 bits : 8 de poids faible dans CH et 2 de poids fort dans les 2 bits forts de CL)

Les cylindres et les têtes sont numérotés à partir de 0 et par conséquent le numéro maximum est n-1. Les numéros de secteurs commencent à 1 et vont jusqu'à n qui en général est 17 en technologie MFM. Les disques à base de technologie RLL commencent à apparaître avec leurs 26 secteurs par piste. En fait, le vrai nombre de piste est supérieur de 1 à celui que vous obtenez mais la piste la plus haute est réservée aux routines de diagnostic et aux tests de maintenance ce qui interdit son accès aux applications.

Faire figurer des marqueurs de mauvais secteurs revient à demander au DOS d'écrire ses *"Bad Sector Markers"* durant le formatage.

Le terme de formatage appliqué au disque dur peut prêter à confusion puisqu'il doit se faire en deux passes avant que l'on puisse utiliser le drive pour y stocker des informations. Le premier formatage, dit hard, est un processus de bas niveau qui écrit les champs d'identificateurs de secteurs (sectors ID) sur le média. Il est en général lancé par une manipulation sous DEBUG. Puis, pour pouvoir utiliser le disque sous MS-DOS, il faut le partitionner (avec l'utilitaire FDISK) et enfin le formater logiquement avec FORMAT. Le formatage logique consiste essentiellement en une vérification rapide du disque afin d'y trouver les zones non lisibles qui sont marquées comme telles dans la FAT (File Allocation Table). Ensuite, MS-DOS installe la FAT et le catalogue sur le disque et reporte la taille du disque et le nombre de mauvais secteurs.

Le formatage hard est normalement un processus d'usine qui doit s'effectuer dans un environnement contrôlé (température, humidité, etc...) qui est à l'origine de la création par le constructeur de la liste des défauts du média et où les pistes défectueuses sont marquées.

Les tentatives des utilisateurs de procéder au formatage hard sont souvent génératrices de difficultés à cause des problèmes de média défectueux où certaines zones apparaissent lisibles au formateur MS-DOS mais qui, plus tard, en cours d'utilisation réelle ne peuvent recevoir certaines données. Cela provoque les *"Unrecoverable Read Error"*, source de bien des soucis. Il est donc préférable de laisser aux experts l'appel du formatage en ROM par le "g=C800:5" sous DEBUG.

2.3.12 Interruption 20 : E/S série

Cette interruption soft fournit les fonctions d'E/S de caractères sur l'un des deux ports série ainsi que les fonctions de configuration de l'interface.

La machine comprend deux ports série : le périphérique série logique 0 (COM1:) qui est toujours configuré et le périphérique série logique 1 (COM2:) optionnel. L'initialisation de mise en route du système détermine si le périphérique série 1 est installé ou pas.

Utilisation des registres du CPU :

Entrée : AH = Numéro de fonction série :

0 - Initialise le port série
1 - Ecriture de caractères vers le port série
2 - Lecture de caractères depuis le port série
3 - Retourne l'état du port série

DX = Numéro de canal logique (0 ou 1)

Sortie : AX = Etat ou caractère retourné selon la fonction choisie

Tous les flags et les autres registres sont préservés.

Si le numéro de canal logique est hors limites (supérieur à 1) ou n'est pas installé, l'opération est abandonnée et le drapeau de timeout (bit 7 de AH) est mis.

Tous les autres bits de AX sont indéterminés.

Les variables compteurs de temporisation du périphérique série (Serial Device Timeout Count) en RAM 0047C et 0047D déterminent le délai d'attente (en demi-secondes) avant l'apparition de l'erreur de timeout. Voir la section 2.4.

Fonction série 0 : Initialisation du port

Cette fonction effectue une ré-initialisation complète d'un canal série. Elle détermine la vitesse de transfert (Baud Rate), le nombre de bits de données (Data bits), le nombre de stop bits et la parité.

Utilisation des registres du CPU :

Entrée : AH = 0
DX = Canal logique (0 ou 1)
AL = Configuration hard :

Bits	Fonction
5 - 7	Code vitesse (0-7). Voir ci-dessous
4	1 = parité paire, 0 = parité impaire
3	1 : contrôle de parité actif
2	1 = 2 stop bits, 0 = 1 stop bit
1	Toujours à 1
0	1 = 8 bits de données, 0 = 7 bits de données

Sortie : AH = registre Line Status du 8250 (voir section 3.4)
AL = registre Modem Status du 8250

Tous les flags et les autres registres sont préservés.

Le code de vitesse de transmission (bits 5 à 7) est ainsi défini :

0 : 110 bauds
1 : 150 bauds
2 : 300 bauds
3 : 600 bauds
4 : 1200 bauds
5 : 2400 bauds
6 : 4800 bauds
7 : 9600 bauds

Si le bit de contrôle du hardware flow de l'octet de mode d'affichage par défaut de la NVR est à 1, RTS est mis vrai et DTR faux. Sinon, l'état courant des lignes est conservé.

Fonction série 1 : Envoi de caractère

Cette fonction provoque l'émission d'un caractère vers le port choisi. Le caractère est émis lorsque les deux status, à savoir CTS et Tx Holding Register Empty du 8250, sont vrais. Si le caractère n'a pu être envoyé durant le délai de timeout spécifié par le compteur en RAM, la commande est abandonnée et le bit 7 de AH est mis à 1.

Utilisation des registres du CPU :

Entrée : AH = 1
AL = Caractère à envoyer
DX = Canal logique (0 ou 1)

Sortie : AH = bits 0 à 6 du registre Line Status du 8250. Si le bit 7 est à 1, le délai d'attente a été dépassé ; sinon, le bit 7 est à 0 et le caractère a pu être envoyé.

A l'appel de cette fonction, RTS est forcé à vrai et si le bit du hardware flow est mis, DTR l'est aussi.

A la sortie, RTS et DTR sont laissés en l'état.

Les variables compteurs de timeout en 0047C et 0047D déterminent le délai d'attente (en demi-secondes) avant l'émission de l'erreur.

Fonction série 2 : Lecture d'un caractère

Cette fonction tente de lire un caractère depuis le port série voulu. Le caractère n'est lu que lorsque les bits d'état de Data Ready (DR) et Data Set Ready (DSR) sont tous les deux vrais. Si aucun caractère n'est reçu durant le délai du compteur timeout, la commande est abandonnée et l'état du timeout modifié en conséquence.

Utilisation des registres du CPU :

Entrée : AH = 2
DX = Numéro de canal logique (0 ou 1)

Sortie : Si un caractère est reçu du 8250, alors :
AL = Caractère reçu
AH = Statut du caractère. Voir tableau ci-dessous :

Bits	Signification
7-5	Toujours à 0
4	Etat du Break
3	Mis à 1 si erreur de framing
2	Mis à 1 si erreur de parité
1	Mis à 1 si erreur d'overrun
0	Toujours à 0

S'il y a erreur de timeout, alors :

AH = 080h (bit 7 = 1)

Dans tous les cas :

Tous les flags et les autres registres sont préservés.

Si le caractère est reçu sans erreur, AH = 0 à la sortie.

A l'entrée, si aucun caractère n'est disponible depuis le port série, DTR (du registre Modem Control) est mis à 1. Lorsqu'un caractère est lu ou lorsqu'il y a timeout, DTR n'est forcé à faux que dans le cas où le contrôle hardware flow est activé.

Si le numéro de canal logique est hors limites (supérieur à 1) ou n'est pas installé, l'opération est abandonnée et le drapeau de timeout (bit 7 de AH) est mis. Tous les autres bits de AX sont indéterminés.

Les variables compteurs de temporisation du périphérique série (Serial Device Timeout Count) en RAM 0047C et 0047D déterminent le délai d'attente (en demi-secondes) avant l'apparition de l'erreur de timeout. Voir la section 2.4.

Fontion série 3 : Demande de l'état du canal

Cette fonction retourne le status du canal logique demandé.

Utilisation des registres du CPU :

Entrée : AH = 3
DX = Numéro de canal logique (0 ou 1)

Sortie : AH = Registre Line Status du 8250 (voir section 3.4)
AL = Registre Modem Status du 8250

Tous les flags et les autres registres sont préservés.

Si le numéro de canal logique est hors limites (supérieur à 1) ou n'est pas installé, l'opération est abandonnée et le drapeau de timeout (bit 7 de AH) est mis. Tous les autres bits de AX sont indéterminés.

2.3.13 Interruption 21 : Fonctions avancées

Cette interruption logicielle donne accès aux particularités hardware avancées de l'Amstrad PC 1640.

Utilisation des registres du CPU :

Entrée : AH = Numéro de fonction avancée :

0 - Lecture et Reset des compteurs X et Y de la souris
1 - Ecriture d'une adresse de la NVR
2 - Lecture d'une adresse de la NVR
6 - Retourne le numéro de version du ROS

Sortie : Si le numéro de fonction est hors limite, alors :
Le Carry est mis
Tous les autres flags sont indéterminés
Tous les registres sont préservés.

Si le numéro de fonction était correct, alors :

L'état des flags et des registres dépend de la fonction.

Fonction avancée 0 : Lecture/Reset des compteurs X/Y de la souris

Cette fonction lit et remet à zéro les registres compteurs X et Y. Chaque registre est lu deux fois. Si les résultats des deux lectures consécutives diffèrent, l'opération est recommencée jusqu'à l'obtention de deux valeurs identiques. A la fin du processus de lecture, les registres sont remis à zéro.

Utilisation des registres du CPU :

Entrée : AH = 0

Sortie : CX = Compteur X signé
DX = Compteur Y signé
Carry = 0

Tous les autres flags sont indéterminés.
Tous les autres registres sont préservés.

Fonction avancée 1 : Ecriture d'une adresse de la NVR

Cette fonction écrit l'adresse de la NVR (Real Time Clock Non-Volatile RAM) demandée, puis recalcule et écrit la somme de contrôle. L'adresse écrite est ensuite relue et comparée à la nouvelle valeur. Un code d'erreur est émis s'il y a différence.

Utilisation des registres du CPU :

Entrée : AH = 1
AL = Emplacement NVR où doit se faire l'écriture (0 à 63)
BL = Donnée NVR à écrire

Sortie : AH = Code de retour =
0 - Ecriture NVR réussie
1 - Emplacement NVR hors-limites
2 - Erreur d'écriture de la donnée NVR
Le Carry est à 0
Tous les autres flags sont indéterminés
Tous les autres registres sont préservés.

Bien que l'on puisse accèder aux adresses 0 à 13 par cette fonction, elles servent au hardware RTC et ne doivent pas être modifiées avec cette fonction.

Vous trouverez l'organisation de la NVR en section 2.5.

Fonction avancée 2 : Lecture d'une adresse NVR

Cette fonction lit l'adresse de la NVR (Real Time Clock Non-Volatile RAM) demandée. La somme de contrôle est calculée, puis comparée avec la valeur réelle et si la checksum NVR est incorrecte, un code d'erreur est retourné.

Utilisation des registres du CPU :

Entrée : AH = 2
AL = Emplacement NVR à lire (0 à 63)

Sortie : AH = Code de retour :
0 - Lecture de la NVR réussie
1 - Emplacement NVR hors-limites
2 - Erreur de checksum de la NVR
AL = Octet lu dans la NVR
Carry false
Tous les autres flags sont indéterminés.
Tous les autres registres sont préservés.

Vous trouverez l'organisation de la NVR en section 2.5.

Fonction avancée 6 : Retourne la version ROS

Cette fonction retourne les deux parties du numéro de version du ROS.

Utilisation des registres du CPU :

Entrée : AH = 6

Sortie : BH = Numéro de Release
BL = Numéro de correction

Carry false
Tous les autres flags sont indéterminés.
Tous les autres registres sont préservés.

Le numéro de release n'est incrémenté que lorsque l'interface avec le ROS est modifiée. Le numéro de correction est changé à chaque version d'une release particulière. Il prend la valeur 0 à chaque nouvelle release.

Remarquez que cette fonction peut être appelée pour déterminer si un programme est exécuté sur un PC Amstrad. Avant l'appel, effacez Carry et mettez BX à 0. Au retour, si Carry = 1 ou si BX = 0, le

programme ne tourne pas sur un Amstrad. Une vérification complémentaire du hardware (sur le port imprimante, voir section 1.10.2) permet de déterminer si la machine est un PC 1512 ou un PC1640.

2.3.14 Interruption 22 : E/S clavier

Cette interruption logicielle donne l'accès au tampon clavier et permet de connaître l'état des diverses bascules.

Utilisation des registres du CPU :

Entrée : AH = Numéro de fonction clavier
- 0 - Retourne un code de touche depuis le tampon clavier
- 1 - Retourne l'état du tampon clavier
- 2 - Retourne l'état des bascules

Sortie : Si numéro de fonction hors limite :
AH = AH-2
Si numéro de fonction correct :
Les registres et flags dépendent de la fonction

Fonction E/S clavier 0 : Demande de code de touche

Retourne le code de touche suivant du tampon clavier. Il y a éventuellement attente de disponibilité d'un code de touche.

Utilisation des registres du CPU :

Entrée : AH = 0

Sortie : AX = Code de touche
Tous les flags et les autres registres sont préservés.

Fonction = E/S clavier 1 : Retourne l'état du tampon clavier

Détermine si le tampon clavier est vide ou non. S'il n'est pas vide, la fonction retourne le code de touche suivant du tampon mais sans l'enlever du tampon.

Utilisation des registres du CPU :

Entrée : AH = 1

Sortie : Si le tampon clavier est vide :
Flag Zero mis
AX indéterminé
S'il y a un code ou plus dans le tampon :
Flag Zero faux
AX = Le prochain code de touche à prendre dans le buffer
Dans tous les cas :
Interruptions rétablies
Tous les autres flags sont indéterminés
Tous les autres registres sont préservés.

Fonction E/S clavier 2 : Retourne l'état des bascules shift

Cette fonction retourne l'état des touches à bascule et des touches alternées à partir de 00417h.

Utilisation des registres du CPU :

Entrée : AH = 2

Sortie : AL = Etat courant :

Bits	Fonction (bit vrai si touche active)
7	INS
6	CAPS LOCK
5	NUM LOCK
4	SCROLL LOCK
3	ALT
2	CTRL
1	SHIFT GAUCHE
0	SHIFT DROIT

Tous les flags et les autres registres sont préservés.

2.3.15 Interruption 23 : E/S Imprimante

Cette interruption logicielle donne accès aux trois canaux d'imprimante.

Utilisation des registres du CPU :

Entrée : AH = Numéro de fonction d'imprimante :

0 - Envoi d'un caractère vers le port imprimante.
1 - Initialisation du port imprimante.
2 - Retourne l'état du port imprimante.

DX = Numéro de canal logique (0 à 2)

L'état des flags et des registres dépend de la fonction.

Sortie : AH = Etat du port imprimante (bits 1 - 7)

Bits	Fonction (bit à 1 si vrai)
7	Imprimante inactive
6	Accusé de réception
5	Fin de papier
4	Sélection de l'imprimante
3	Erreur d'E/S
1 et 2	Toujours 0
0	0 si E/S réussie ou 1 si Timeout (voir les fonctions d'imprimante)

Tous les flags et les autres registres sont préservés.

Cette fonction donne accès à trois canaux logiques. Le canal 0 est le port système et il est standard sur toutes les machines. La séquence d'initialisation de mise en route détermine s'il y a d'autres ports parallèles externes installés. Lorsque les deux

interfaces d'imprimante additionnelles sont présentes, le device 1 est le port d'imprimante externe et device 2 le port de la carte vidéo monochrome. Une seule interface additionnelle est montée, elle est toujours le device logique 1.

Les variables compteurs de timeout des imprimantes se trouvent en 0478h - 047Ah (voir section 2.4).

Fonction imprimante 0 : Ecrit un caractère

Cette fonction tente d'envoyer un caractère vers le port imprimante désigné. Si ce caractère ne peut être envoyé dans le délai spécifié par la variable compteur de timeout, l'opération est abandonnée et AH est retourné avec le bit 0 mis à 1.

Utilisation des registres du CPU :

Entrée : AH = 0
AL = Caractère à envoyer
DX = Numéro de canal logique (0 à 2)

Sortie : AH = Etat du port imprimante (voir tableau précédent) ou bit de Timeout (0) mis.
Tous les flags et les autres registres sont préservés.

Fonction imprimante 1 : Initialisation du canal imprimante

Cette fonction exécute une ré-initialisation complète du canal spécifié (s'il est présent). Le signal d'imprimante INIT est maintenu bas durant environ 4 millisecondes. Les interruptions d'imprimante et l'auto line feed sont inhibés.

Utilisation des registres du CPU :

Entrée : AH = 1
DX = Canal logique (0 à 2)

Sortie : AH = Etat du port imprimante ou bit de canal invalide (0) allumé.
Tous les flags et les autres registres sont préservés.

Fonction d'imprimante 2 : Demande d'état du canal

Cette fonction retourne le registre d'état du canal d'imprimante spécifié (s'il existe).

Utilisation des registres du CPU :

Entrée : AH = 2
DX = Numéro de canal logique (0 à 2)

Sortie : AH = Etat du port imprimante ou bit de canal invalide (Bit 0) mis à 1.
Tous les flags et les autres registres sont préservés.

2.3.16 Interruption 24 : System Restart

SI une ROM BIOS de disque dur a été initialisée à la mise en route de la machine, elle aura installé son propre point d'entrée dans ce vecteur et sauvegardé le vecteur ROS dans sa zone de variables. Son processus de bootstrap ressemble à celui décrit dans la section suivante. S'il se déroule bien, le système sera chargé à partir de la partition active du disque. Sinon, il y a lancement de l'Int 24 du ROS dont le vecteur a été sauvegardé.

Cette interruption logicielle permet de lancer un redémarrage du système. Un message demandant à l'utilisateur d'insérer une disquette système dans le lecteur A et de presser une touche quelconque s'inscrit sur le périphérique d'affichage actif. Quand la pression de la touche est reçue, le processus de disk bootstrap (Int 25) est appelé.

Utilisation des registres du CPU :

Entrée : pas de condition

Sortie : Disk Bootstrap

2.3.17 Interruption 25 : Disk Bootstrap

Cette interruption logicielle donne accès au processus de bootstrap qui a normalement lieu après la fin des tests de mise en route.

Le ROS tente de charger le boot secteur (lecteur A, face 0, piste 0, secteur 1) vers l'adresse 07C00h. Si le chargement se déroule avec succès, le programme du secteur prend le contrôle grâce à un saut long vers le segment 0000, offset 7C00. Si le bootstrap secteur ne peut être chargé après 10 tentatives, le ROS affiche un message *"Insert a SYSTEM disk into drive A"* et *"Then press any key"*. Le ROS attend alors l'appui d'une touche et répète la procédure de redémarrage du système (Int 24).

L'appel de cette interruption peut ne pas marcher sous DOS 3.2 si un processus quelconque a été installé en mémoire de manière à ce qu'il soit connecté à une interruption active quelconque telle que l'interruption ticker de l'horloge système. Le ROS effectue le bootstrap sans initialiser les vecteurs d'interruption et si une routine était résidente sous MS DOS, elle sera effacée durant le redémarrage mais son appel par interruption sera toujours présent. Le contrôle sera donc passé à un moment ou à un autre à l'adresse où résidait précédemment la routine ce qui bloquera le système puisque le processeur va se mettre à exécuter un code qui n'en est pas un. Les programmes tels que MOUSE.COM entrent dans cette catégorie de processus résidents appelés depuis l'interruption d'horloge.

Utilisation des registres du CPU :

Entrée : Pas de condition

Sortie : Vers le programme chargé par le bootstrap

2.3.18 Interruption 26 : Horloge système et horloge temps réel

Cette interruption logicielle donne accès à la fois à l'adresse de l'horloge système (maintenue par logiciel) et à l'horloge temps réel (RTC) hardware.

Utilisation des registres du CPU :

Entrée : AH = Spécificateur de fonction d'horloge :

0 - Lit l'horloge système
1 - Modifie l'horloge système
2 - Lit l'heure RTC
3 - Règle l'heure RTC
4 - Lit la date RTC
5 - Fixe la date RTC
6 - Règle l'alarme RTC
7 - Annule l'alarme RTC

Fonction horloge 0 : Lit l'horloge système

Cette fonction retourne la valeur 32 bits de l'horloge système.

Utilisation des registres du CPU :

Entrée : AH = 0

Sortie = DX = Mot de poids faible du compteur d'horloge
CX = Mot de poids fort du compteur d'horloge
AL = Flag 24 heures :
0 si 24 heures ne se sont pas passées
1 si plus de 24 heures écoulées

Tous les flags et les autres registres sont préservés.

L'horloge système sur 32 bits est incrémentée toutes les 54,9 millisecondes par la routine d'interruption hardware de la System Clock (ticker). Lorsque le compteur atteint la valeur des 24 heures (001800B0h), le flag des 24 heures est mis et le comptage repart de 0. Ce comptage sur 24 heures est basé sur la fréquence de l'horloge système 1.19318 Mhz divisée par le plus grand diviseur : 65536. Cela donne une interruption toutes les 54.92549323 Ms qui, divisé par le nombre de secondes en 24 heures donne la valeur indiquée plus haut.

Notez que le drapeau 24 heures est remis à 0 après lecture.

Fonction horloge 2 : Lit l'heure RTC

Cette fonction lit l'heure de l'horloge temps réel.

Utilisation des registres du CPU :

Entrée : AH = 2

Sortie : Si RTC ne fonctionne pas :
Carry vrai
CX et DX préservés.
Si RTC fonctionne :
Carry faux
CH = Heures en décimal codé binaire (BCD)
CL = Minutes (BCD)
DH = secondes (BCD)
Dans tous les cas :
Tous les autres flags sont indéterminés.
Tous les autres registres sont préservés.

Fonction horloge 3 : Règle l'heure RTC

Cette fonction détermine l'heure de l'horloge temps réel.

Utilisation des registres du CPU :

Entrée : AH = 3
CH = Heures (BCD)
CL = Minutes (BCD)
DH = Secondes (BCD)
DL = 1 pour activer l'option heure d'été/heure d'hiver,
0 sinon

Sortie : Si RTC ne fonctionne pas :
Carry vrai
Si RTC fonctionne :
Carry faux
Dans tous les cas :
Tous les autres flags sont indéterminés.
Tous les autres registres sont préservés.

Quand l'option heure d'été/heure d'hiver est active, l'horloge fonctionne de la façon suivante : le dernier dimanche d'Avril, l'incrémentation de l'heure se fait de 1:59:59 AM (c'est-à-dire du matin) à 3:00:00 AM. Le dernier dimanche d'Octobre, l'heure passe de 1:59:50 AM à 1:00:00 AM.

Notez que cette fonction inhibe aussi l'alarme.

Fonction horloge 4 : Donne la date RTC

Cette fonction lit la date de l'horloge temps réel.

Utilisation des registres du CPU :

Entrée : AH = 4

Sortie : Si RTC ne fonctionne pas :
Carry vrai
CX et DX préservés.
Si RTC fonctionne :
Carry faux
CH = Siècle en décimal codé binaire (BCD)
CL = Année (BCD)
DH = Mois (BCD)
DL = Jour (BCD)
Dans tous les cas :
Tous les autres flags sont indéterminés.
Tous les autres registres sont préservés.

L'octet du siècle est mis à 19 (BCD) si l'année est supérieure ou égale à 80 (BCD), autrement, il est mis à 20 (BCD).

Fonction horloge 5 : Règle la date RTC

Cette fonction fixe la date de l'horloge temps réel.

Utilisation des registres du CPU :

Entrée : AH = 5
CH = Siècle en décimal codé binaire (BCD)
CL = Année (BCD)
DH = Mois (BCD)
DL = Jour (BCD)

Sortie : Si RTC ne fonctionne pas :
Carry vrai
Si RTC fonctionne :
Carry faux
Dans tous les cas :
Tous les autres flags sont indéterminés.
Tous les autres registres sont préservés.

Le siècle est ignoré et est calculé comme indiqué dans la description de la fonction 4.

Fonction horloge 6 : Règle l'alarme RTC

Cette fonction fixe l'heure d'alarme et active l'interruption d'alarme RTC. Cette interruption d'alarme interviendra lorsque l'heure courante atteindra l'heure d'alarme. Une application utilisant cette fonction doit tout d'abord écrire l'adresse de sa routine d'interruption d'alarme dans le vecteur d'interruption 10.

Utilisation des registres du CPU :

Entrée : AH = 6
CH = Heures (BCD)
CL = Minutes (BCD)
DH = Secondes (BCD)

Sortie : Si l'alarme RTC est déjà active :
Carry vrai
Sinon :
Carry faux

Dans tous les cas :
Tous les autres flags sont indéterminés.
Tous registres sont préservés.

Fonction horloge 7 : Annule l'alarme

Cette fonction désarme la fonction d'alarme de la RTC.

Utilisation des registres du CPU :

Entrée : AH = 7

Sortie : Tous les flags et les registres sont préservés.

2.3.19 Interruption 27 : Break clavier

Cette interruption logicielle est appelée par la routine d'interruption hard du clavier quand ce dernier détecte la combinaison [Ctrl]+[Num Lock].

L'initialisation de mise en route charge l'adresse d'une routine contrôleur de break fictive qui ne se compose que d'une instruction de retour d'interruption IRET.

Les programmes d'application qui traitent cette interruption doivent se conformer aux règles suivantes :

Entrée : DS = 0040h (passe par dessus les données du ROS)

Sortie : Tous les registres doivent être préservés sauf AX, BX, CX, DX et DS. Les flags peuvent être altérés.

La routine d'interruption fournie ne doit pas faire appel à d'autres interruptions du ROS mais peut modifier toutes les adresses système en RAM du ROS.

2.3.20 Interruption 28 : Interruption Ticker externe

Cette interruption logicielle est appelée depuis la routine d'interruption hardware de l'horloge système. Elle est initialisée avec une routine fictive qui ne comprend qu'une instruction IRET (retour d'interruption) lors de la mise en route de la machine. Elle peut être utilisée par une application qui aurait besoin de lancer une opération à intervalles réguliers.

Les programmes d'application qui traitent cette interruption doivent se conformer aux règles suivantes :

Entrée : DS = 0040h (passe par dessus les données du ROS)

Sortie : Tous les registres doivent être préservés sauf AX, DX et DS. Les flags peuvent être altérés.

La routine d'interruption fournie ne doit pas faire appel à d'autres interruptions du ROS mais peut modifier toutes les adresses système en RAM du ROS.

2.3.21 Interruption 29 : Table des paramètres vidéo

L'adresse de ce vecteur d'interruption renferme l'adresse sur 20 bits de la table des paramètres vidéo qui sert à la configuration du MC6845 (CRTC) lors du changement de mode vidéo. A la mise en route ou après un reset, le ROS affecte l'adresse de cette table à ce vecteur situé en 0074-0077h.

La table des paramètres vidéo se compose de 4 entrées consécutives de 16 bits. Chaque entrée contient une valeur d'initialisation pour l'un des registres du MC6845 (voir section 1.11.10). Lorsque l'on sélectionne un nouveau mode vidéo, l'entrée de la table qui sert à l'initialisation est la suivante :

Entrée	Mode vidéo
0	0 - Texte 25 lignes x 40 caractères 1 - Texte 25 lignes x 40 caractères
1	2 - Texte 25 lignes x 80 caractères 3 - Texte 25 lignes x 80 caractères
2	4 - Graphique 320 x 200, palettes 0 ou 1 5 - Graphique 320 x 200, palette 2 6 - Graphique 640 x 200
3	7 - Texte 25 x 80 caractères, carte Monochrome

La table contient les paramètres d'initialisation suivants :

Numéro de reg.	Fonction	Entrée 0	Entrée 1	Entrée 2	Entrée 3
R0	Balayage total horizontal	56	113	56	97
R1	Affichage horizontal	40	80	40	80
R2	Position synchro hori.	45	90	45	82
R3	Synchro largeur hori.	10	10	10	15
R4	Balayage total vertical	31	31	127	25
R5	Ajustage vertical	06	06	06	06
R6	Affichage vertical	25	25	100	25
R7	Position synchro vert.	28	28	112	25
R8	Interlaçage	02	02	02	02
R9	Nbre lignes max/caract.	07	07	01	13
R10	Départ curseur	06	06	06	11
R11	Arrivée curseur	07	07	07	12
R12	Départ écran fort	00	00	00	00
R13	Départ écran faible	00	00	00	00
R14	Position curseur fort	00	00	00	00
R15	Position curseur faible	00	00	00	00

2.3.22 Interruption 30 : Table des paramètres disque

L'adresse de ce vecteur contient l'adresse 32 bits de la table des paramètres de configuration de l'interface disques. A la mise en route ou après un reset, le processus d'initialisation charge l'adresse de la table en ROM dans cette adresse de vecteur (0078 - 007Bh).

La table de paramètres disque se compose de 11 octets :

Octet	Fonction	Val. hexa
0	2ième octet de commande au contrôleur (Step Rate : 6 ms, délai Head Unload :15)	DF
1	3ième octet de commande au contrôleur (délai Head Load : 1, contrôleur FDC DMA = 0)	02
2	Timeout arrêt moteur (environ 5,4 s)	64
3	Taille secteur (512 octets)	02
4	End of Track (secteur 9)	09
5	Gap Length en lecture/écriture	2A
6	DTL - Longueur des données	FF
7	Gap Length au formatage	50
8	Octet de remplissage au formatage	F6
9	Délai Head Setting (15 ms)	0F
10	Délai mise en route moteur (500 ms)	04

2.3.23 Interruption 31 : Table matrice vidéo (Video Matrix Table)

L'adresse de ce vecteur d'interruption renferme l'adresse 32 bits de table de matrice vidéo qui sert dans les modes graphiques compatibles à générer les pixels des caractères 128 à 255.

A la mise en route ou après un reset, le processus d'interruption écrit des 0 dans cette adresse (007C-007Fh) pour indiquer qu'aucune matrice externe n'est chargée. Les programmes tels que GRAFTABL.EXE chargent une matrice résidente de 128 caractères que l'on peut utiliser en modes graphiques 640x200.

De plus, la ROM BIOS IGA se sert d'un autre vecteur, le n° 67 (adresse en 0:10Ch) qui est initialisé de manière à ce qu'il pointe vers une table en ROM complète de 256 caractères. La fonction 17 (sous-fonction 48, BH = 4) de l'interruption Vidéo IGA 16 peut servir à obtenir l'adresse d'origine de la matrice 8x8 en ROM des 128 caractères supérieurs. La valeur ainsi obtenue peut venir occuper ce vecteur d'interruption 31.

Lorsque les jeux de caractères étrangers de la ROM de fontes IGA étendue sont utilisés (sw9 ON), une table de vecteurs est disponible en C7FE8 (C000:7FE8). Elle se compose de 12 entrées (mots) dont les quatre premiers sont pour les caractères danois, les quatre suivants pour les caractères Portugais et les quatre derniers pour les caractères grecs. Le premier pointeur dans chacun de ces groupes est pour le jeu des 128 caractères 8x8 supérieurs de la variante nationale. En se servant de la valeur d'offset avec un segment égal à C000h, on peut obtenir les jeux de 128 caractères supérieurs des variantes nationales sans avoir à charger de matrices sur disque.

Chacune des 128 entrées de la table de fontes se compose de 8 octets qui forment chacun un ligne de pixels (8x1) du caractère. Le premier octet correspond à la valeur de la ligne supérieure et le dernier octet à la valeur de la ligne de base du caractère. Le bit de poids fort (bit 7) est le pixel de gauche et le bit de poids faible le pixel de droite. Lorsqu'un bit est à 1, le pixel prend la couleur d'avant-plan, sinon le pixel prend la couleur du fond.

2.4 VARIABLES SYSTEME EN RAM

La zone des variables système servant au ROS s'étend de 00300 à 00500. Le tableau suivant donne la liste des variables et leur utilisation. Elles sont classées en Octets (8 bits), Mots (16 bits), Mot Long (32 bits) ou Tampons (ou Buffer, de plus de 32 bits). Selon la manière dont on charge les registres de segment, on peut considérer que, par exemple, les variables à 4XX se trouvent à 40:XX ou à 0:4XX.

Adresse(s)	Usage
0300 à 03FF	Pile d'initialisation (Tampon) Sert comme zone de pile durant l'initialisation
00400	Logical Serial Device 0 Base I/O Address (Mot) Contient l'adresse de base du périphérique série logique 0. D'origine, l'adresse du port série asynchrone du système.
00402	Logical Serial Device 1 Base I/O Address (Mot) Contient l'adresse de base du périphérique série logique 1. D'origine, le port série asynchrone externe ou zéro s'il n'est pas installé.
0404 à 0407	Réservé.
00408	Logical Printer Device 0 Base I/O Address (Mot) L'adresse de base de l'imprimante logique 0. D'origine, le port parallèle du système.
0040A	Logical Printer Device 1 Base I/O Address (Mot) L'adresse de base de l'imprimante logique 1. S'il est présent, c'est le port parallèle externe, sinon c'est celui de la carte monochrome externe. Si aucun de ces deux devices n'est installé, l'adresse reçoit 0.
0040C	Logical Printer Device 2 Base I/O Address (Mot) L'adresse de base de l'imprimante logique 2. D'origine, pointe vers la carte monochrome externe si elle est installée et s'il y a déjà un port parallèle externe. Sinon, prend la valeur 0.
0040E	Réservé (Mot)

00410

System Configuration Status (Mot)
Contient la configuration système :

Bits	Fonction
14 et 15	Nombre d'imprimantes (1 à 3)
13	Inutilisé
12	Mis à 1 si manette de jeux optionnelle
11	Toujours 0
9 et 10	Nombre d'interface série (1 ou 2)
8	Inutilisé
7	Toujours 0
6	Mis à 1 si second floppy
4 et 5	Mode d'affichage par défaut (DDM)
2 et 3	Toujours 1
1	Mis à 1 si 8087 installé
0	Toujours 1

00412

Réservé (Octet)

00413

Total RAM Size (Mot)
Nombre de blocs de 1 K de RAM utilisateur (du système) installés. 640 pour un PC 1640.

00415

Extra RAM Size (Mot)
Reçoit à l'initialisation le nombre de blocs de 1 K de mémoire utilisateur (du système) intallée moins 64 (donc 576 pour un PC 1640).

00417

Key Toggles and Key States (Octet)
Cet octet sert à garder trace de l'état des bascules du clavier (bits 4 à 7) et des touches alternées (bits 0 à 3) :

Bit	Touche (1 si active)
7	INS
6	CAPS LOCK
5	NUM LOCK
4	SCROLL LOCK
3	ALT
2	CTRL
1	SHIFT GAUCHE
0	SHIFT DROITE

00418

Key down (Octet)
Cet octet sert à enregistrer l'état des touches bascules de manière à ce qu'elles ne soient pas répétées lorsqu'on tient une touche enfoncée.

Bit	Touche (1 si enfoncée)
7	INS
6	CAPS LOCK
5	NUM LOCK
4	SCROLL LOCK

Le bit 3 est mis à 1 lorsque l'état Pause ([Ctrl][Num Lock]) est actif. Les bits 0 à 2 sont inutilisés et initialisés à 0.

00419

Absolute Key Token Number (Octet)
Lorsqu'un code touche absolu est entré via la touche ALT et le pavé numérique, cette variable reçoit l'état courant du code.

0041A	Key Token Buffer Out Pointer (Mot) Cette variable renferme l'offset absolu vers le prochain code de touche à prélever dans le tampon des codes de touches. Notez que pour le ROS, ce tampon a une adresse de départ de segment égale à 0040h.
0041C	Key Token Buffer In Pointer (Mot) Cette variable renferme l'offset absolu du prochain emplacement vide du tampon des codes touches. Lorsque cette adresse est la même que le Out Pointer, le tampon est vide.
0041E	Key Token Buffer (Tampon) Le tampon des codes touches est un tampon FIFO (circulaire) qui permet de stocker 16 codes de touches du clavier.
0043E	Drive Restore Flag (Octet) Chaque lecteur de disquette a un flag "Restore" (bit 0 pour le drive 0, bit 1 pour le drive 1). Si cet indicateur est à 0 avant un accès disque quelconque (lecture, écriture, vérification, formatage), la commande restore est émise vers le FDC pour le lecteur concerné. Si l'opération se déroule avec succès, le bit correspondant est mis à 1. A l'appel de la fonction d'initialisation de l'interruption disque, le flag "Restore" est effacé.Le bit 7 sert au traitement des interruptions hardware du contrôleur de disquettes (FDC).
0043F	Drive Motor Flag (Octet) Lorsque le moteur d'un lecteur de disquette tourne, le bit correspondant au lecteur (0 pour le drive 0, 1 pour le drive 1) est mis à 1.

00440	Drive Motor Timeout Counter (Octet) Après chaque opération sur les disques, le compteur de temporisation d'arrêt du moteur est chargé avec la valeur trouvée dans la table des paramètres disques (voir interruption 30). A chaque éxécution de l'interruption d'horloge système, le compteur est décrémenté. Quand il atteint 0, le Flag Drive Motor est mis à 0.
00441	Disk Status (Octet) Cet octet reçoit l'état retourné par la dernière opération disque (voir section 2.3.11, Interruption E/S disques, fonction 1).
00442	FDC Results Buffer/HD Parameter Buffer (Tampon) Ce tampon de 7 octets conserve les informations relatives à l'état du contrôleur de disquettes retourné à la fin de chaque opération sur disquettes. Il sert aussi au stockage de paramètres d'appel d'opérations d'E/S de la ROM BIOS disque dur.
00449	Current Video Mode (Octet) Le mode vidéo sélectionné par le dernier appel de la routine appropriée de Int 16 est rangé ici.
0044A	Visible Display Columns (Mot) Nombre de colonnes de caractères visibles.
0044C	Video Display Page Size (Mot) Ce mot représente la quantité de RAM vidéo servant à l'affichage d'une page par la ROM BIOS. Cette valeur est représentée comme suit :

Mode(s)	Taille
0 et 1	2048
2 et 3	4096
4 à 6	16384
7	4096
13	8192
14	16384
15 et 16	32768

0044E — Display Page Start Offset (Mot)
Renferme l'offset de la page vidéo actuellement active.

00450 — Cursor Address Buffer (Tampon)
Ce tampon de 16 octets contient l'emplacement du curseur en lignes et colonnes de 8 pages d'écran maximum. Ce nombre correspond donc au nombre de pages maximum que puissent traiter les routines vidéo de la ROM BIOS.

00460 — Cursor End Scan (Octet)
Cet octet représente le numéro de ligne (de balayage) à l'intérieur de la matrice d'un caractère d'arrivée du curseur. Cette valeur sert à la programmation du contrôleur CRT.

00461 — Cursor Start Scan (Octet)
Cet octet représente le numéro de la ligne (de balayage) de la matrice d'un caractère où se situe le départ du curseur. Cette valeur sert à la programmation du contrôleur CRT.

00462 — Active Display Page (Octet)
Numéro de la page écran active.

00463	CRTC I/O Address (Mot) Adresse d'E/S de l'interface CRTC (contrôleur vidéo) actuellement utilisée (Mono : 3B4, Couleur : 3D4).
00465	Current Video Mode Control Register (Octet) Cet octet est la copie de la valeur du registre Video Mode Control.
00466	Current CGA Color Select Register (Octet) Cet octet est la copie de la valeur du registre de séléction de couleur CGA.
0467 à 046B	Réservé
0046C	System Clock (Mot Long) Le compteur 32 bits de l'horloge système.
00470	24 Hour Flag (Octet) Lorsque l'horloge système atteint 0001800B0h, elle prend la valeur 0 et cet octet est mis à 0FFh. Notez que la lecture de l'horloge système par l'interruption 26 remet ce flag à 0.
00471	Break (Octet) Initialement, cet octet prend la valeur 0. Chaque fois que la combinaison de touches [Ctrl][Num Lock] (qui correspond à Break) est détectée, le bit 7 est mis à 1. Une application qui se sert de ce bit pour détecter le Break doit le remettre à 0 après lecture.

00472	System Reset Flag (Mot) Lorsqu'un system reset est détecté ([Ctrl][Alt][Del]), cette adresse prend la valeur 01234h avant que ne soit effectivement lancé le reset. Cela permet à la routine de mise en route - qui vient lire ce flag - de savoir qu'elle ne doit pas recommencer les tests de la RAM. Donner la valeur 1235h à cette adresse interdit un reset hardware complet.
00474	Last Hard Disk I/O Completion Code (Octet) La ROM BIOS disque dur range à cette adresse le code de retour de la dernière opération d'E/S sur le disque dur.
00475	Hard Disk Drive Count (Octet) Cette adresse sert à la ROM BIOS disque dur à conserver le nombre de disques durs connectés.
0476 et 0477	Réservé à la ROM BIOS disque dur.
0478 à 047A	Logical Printer Device 0 - 2 Timeout Count (3 octets) Ces compteurs de temporisation déterminent le temps (en demi-secondes) durant lequel le ROS va essayer d'envoyer un caractère à un canal logique d'imprimante. Initialement, ils ont la valeur 20 (10 secondes).
0047B	Réservé.
047C et 047D	Logical Serial Device 0 - 1 Timeout Cont (2 octets) Ces compteurs de temporisation déterminent le temps d'attente (en demi-secondes) pour les opérations d'E/S avec un canal série logique particulier. Ils sont initialisés pour une attente d'une seconde.

0047E	Réservé.
00480	Key Token Buffer Start Address (Mot) Pointeur offset vers le début du tampon des codes clavier. Notez que l'adresse implicite de segment est 0040h.
00482	Key Token Buffer End Address (Mot) Pointeur offset vers la fin du tampon des codes de clavier.
00484	Display Rows (Octet) Cette adresse contient le nombre de lignes de caractères moins une de l'affichage.
00485	Character Points (Octet) Cette adresse contient la longueur (en octets) de la matrice de caractères courante.
00487	EGC Status (Octet) Cette adresse sert à la ROM BIOS IGA pour y conserver sa valeur d'état courant (Color/Mono, Primary/Sec).
00488	Switches (Octet) Cette adresse indique la configuration des switches 1 à 4 de l'EGC dans les bits 0 à 3 (inversés). Les quatre bits supérieurs sont à 1

<table>
<tr><td>00500</td><td>Print Screen Status (Octet)
Cet octet reflète l'état de l'opération d'impression d'écran :

<table><tr><th>Valeur</th><th>Signification</th></tr><tr><td>0</td><td>Impression achevée sans erreur.</td></tr><tr><td>1</td><td>Impression en cours.</td></tr><tr><td>255</td><td>Impression abandonnée (timeout).</td></tr></table></td></tr>
</table>

2.5 RAM NON VOLATILE (NVR pour Non Volatile RAM)

Les 40 premiers octets de la RAM préservée par piles dans le hardware RTC servent au paramètrage du système :

Octets	Usage	Défaut
0 à 9	Paramètres date et heure RTC	--
1	Registre de contrôle A de la RTC	070h
11	Registre de contrôle B de la RTC	002h
12	Registre de contrôle C de la RTC	--
13	Registre de contrôle D de la RTC	--
14 à 19	Date et heure de la dernière utilisation de la machine	--
20	Checksum RAM utilisateur	
21 et 22	Code de transposition de la touche Enter	01C0Dh
23 et 24	Code de transposition de la touche Delete en avant	02207h
25 et 26	Code de transposition du bouton Feu 1 du joystick	0FFFFh

Octets	Usage	Défaut
27 et 28	Code de transposition du bouton Feu 2 du joystick	0FFFFh
29 et 30	Code de transposition du bouton 1 de la souris	0FFFFh
31 et 32	Code de transposition du bouton 2 de la souris	0FFFFh
33	Facteur du déplacement en X de la souris	00Ah
34	Facteur du déplacement en Y de la souris	00Ah
35	Mode vidéo initial & nbre de drives	020h
36	Attribut initial des carac. vidéo	007h
37	Taille du RAM disk en blocs de 2 K	000h
38	Octet de configuration initiale de l'UART interne	0E3h
39	Octet de configuration initiale de l'UART externe	0E3h
40 à 63	Inutilisés.	Zéros

Les octets 0 à 13 sont les registres hardware de l'horloge temps réel. Reportez-vous à la section 3.8 où vous trouverez l'explication de leur utilisation et de leur valeur.

Le processus d'initialisation de mise en route (à la mise sous-tension ou après un reset) comprend une phase de calcul d'une somme de contrôle de la NVR. Si l'octet de poids faible de la somme n'est pas 0AAh ou si le bit de faible voltage de la pile est à 1 (bit du registre d'état de la NVR), les valeurs par défaut sont affectées à ces variables NVR et un message d'avertissement est émis vers le périphérique d'affichage principal. Les emplacements où, dans le tableau ci-dessus, ne figurent pas de valeurs par défaut (marqués —) ne sont pas modifiés.

Le code touche par défaut des octets 25 à 32 est la valeur spéciale 0FFFFh qui indique à l'interruption hardware du clavier d'ignorer l'appui de la touche plutôt que d'insérer le code dans le tampon.

Le mode vidéo initial (octet 35) sert aux différentes affectations de valeurs que fait le ROS durant l'initialisation de mise en route et dépend de la configuration des switches. Il sert ensuite à l'initialisation du canal status-1 (DDM - bits 4 et 5. Voir les combinaisons valides en 1.8.2). Les bits 4 et 5 de l'octet 35 prennent leur valeur en rapport. Le bit 6 est mis à 1 si deux lecteurs sont montés, sinon il vaut 0. L'octet 35, en mode par défaut, a le bit 6 mis à 1 (2 lecteurs) et les bits 4 et 5 valant respectivement 1 et 0 (Couleur, texte, 80 caractères x 25 lignes).

Le bit 7 de l'octet 35 active ou désactive l'option flow control des E/S série. Voir la section 2.3.12.

Lorsque l'on se trouve en mode texte, l'attribut de caractère initial (octet 36) est recopié vers tous les octets d'attribut du tampon RAM vidéo. La valeur par défaut donne un avant-plan blanc sur un fond noir.

La taille du RAM disk (octet 37) set à indiquer la taille du disque virtuel initial à MS-DOS.

Les paramètres UART initiaux (octets 38 et 39) sont définis ainsi : 9600 baud, 8 bits de données, 1 bit d'arrêt et pas de parité. Ces valeurs sont affectées à leur canal série respectif par la fonction d'initialisation série (Interruption 20. Voir section 2.3.12).

2.6 LES MESSAGES DU ROS

Cette section recense les messages émis (éventuellement) par le ROS durant l'initialisation de mise en route et les autotests. Le langage dans lequel sont émis ces messages dépend des trois liaisons optionnelles LK1 - LK3 reflétées par les 3 bits de poids faible du Printer Status Channel (voir le détail en section 3.1).

2.6.1 Messages d'information ou d'avertissement.

Les messages suivants servent soit à donner un renseignement sur le déroulement des opérations, soit à mettre en garde l'utilisateur

contre un problème potentiel. Ils n'indiquent pas l'occurrence d'une erreur fatale et le processus d'initialisation se poursuit même dans le cas où un test aurait échoué.

Ces messages sont émis vers l'écran primaire (dans le mode d'affichage par défaut tel que défini dans la NVR).

Pease wait ou Patientez

Ce message est affiché sur la première ligne de l'écran après une mise sous-tension ou un system reset au clavier ([Ctrl][Alt][Del]). A chaque fois qu'un autotest particulier s'est achevé de manière satisfaisante, un nouveau point s'affiche sur la deuxième ligne de l'écran.

AMSTRAD PC nnnK (r.c) Last used at hh:mn on jj mm aa
AMSTRAD PC nnnK (r.c) hh:mn le jj mm aa

Ce message apparaît après que tous les tests aient été passés avec succès. Dans celui-ci :

nnn = Taille mémoire vive en kilo-octets.

r.c = numéro de release (r) du ROS et numéro de correction (c).

hh:mn = heure (hh) et minute (mn) de la dernière utilisation.

jj mm aa = jour (jj), mois (mm) et année (aa) de la dernière utilisation.

Please fit new batteries (Changez les piles SVP)

Ce message vient s'inscrire sous le précédent lorsque le ROS a détecté que le bit de faible voltage de la pile de la RTC était à 1. Cela indique l'absence ou la défaillance proche des piles.

Check keyboard and mouse (Vérifiez le clavier et la souris)

Ce message indique que la valeur retournée par l'autotest du clavier était différente du message de bon fonctionnement (0AAh).

Insert a SYSTEM disk into drive A
Then press any key (Insérez une disquette système dans le lecteur A et appuyez sur une touche)

Ce couple de messages survient lorsque le boot secteur de la disquette n'a pu être lu dans le drive A après 10 tentatives.

Erro :External ROM checksum incorrect:ROM address=nnnnnh
(Erreur : Somme de contrôle d'une ROM externe incorrecte : Adresse de la ROM = nnnnnh)

Ce message indique que la checksum d'une ROM externe était différente de 0 (voir la section 2.1.16). L'adresse physique de la ROM est indiquée sous la forme d'un nombre hexadécimal à 5 chiffres.

2.6.2 Messages d'erreurs fatales

Les messages suivants sont le signe qu'un autotest particulier a échoué et que l'initialisation ne peut se poursuivre. Dans un tel cas, il faut éteindre la machine puis la rallumer pour qu'elle recommence le cycle d'initialisation. Le mode d'affichage passe en mode texte 8O colonnes avant l'émission de ces messages.

Error:Faulty SYSTEM RAM
RAM du système défectueuse

Error:Faulty VDU RAM
RAM VDU défectueuse

Error:Faulty interrupt controller
Contrôleur d'interruptions défectueux

Error:Faulty direct memory access controller
Contrôleur DMA défectueux

Error:Faulty floppy disk controller or disk drive
Contrôleur de lecteurs de disquette ou lecteur de disquette défectueux

Error:Faulty interval timer
Timer défectueux

Error:Faulty system status register
Erreur de registre d'état du système

Error:Faulty real time clock
Horloge temps réel défectueuse

Error:Faulty VDU controler
Contrôleur VDU défectueux

Error:Faulty system printer port
Port imprimante du système défectueux

Error:Faulty system serial port
Port série du système défectueux

Error:Faulty faulty mouse coordonate register
Erreur de registre de coordonnées de la souris

Error:Faulty ROS ROM checksum
Erreur de checksum de la ROM ROS

Error:Faulty memory (parity error)
Erreur de parité de la mémoire

Lorsque l'un de ces tests échoue, les suivants ne peuvent avoir lieu car il se peut qu'ils fassent appel à l'élément défectueux. C'est pourquoi le système entre dans une boucle infinie et non interruptible. Une erreur de cette sorte ne devrait normalement pas intervenir même épisodiquement.

CHAPITRE 3

TABLE DE REFERENCE

Les tableaux suivants présentent certaines caractéristiques matérielles ou logicielles de l'Amstrad PC 1640 dont certaines peuvent avoir été mentionnées précédemment dans d'autres chapitres mais qui figurent tout de même ici pour que vous puissiez trouver facilement l'information qui vous intéresse.

3.1 LES LIAISONS DEFINISSANT LE LANGAGE

Les 3 bits de poids faible du Printer Status Channel (adresse d'E/S 379) sont le reflet (en complément à 1) de l'état du groupe des liaisons optionnelles LK1-LK3 (situées sur le côté gauche de la carte mère à environ 5 centimètres sous le connecteur d'imprimante). Ils servent à indiquer au système d'exploitation du ROS la langue choisie pour ses messages ou l'option du mode diagnostic.

LK1	LK2	LK3	Correspondance
OFF	OFF	OFF	Anglais
OFF	OFF	ON	Allemand
OFF	ON	OFF	Français
OFF	ON	ON	Espagnol
ON	OFF	OFF	Danois
ON	OFF	ON	Suèdois
ON	ON	OFF	Italien
ON	ON	ON	MODE DIAGNOSTIC

Les messages du ROS sont émis dans la langue sélectionnée. En mode diagnostic, les messages sont de nouveau affichés en anglais et les tests normaux sont effectués mais si l'un d'entre eux échoue, le

processus se poursuit tout de même et un bootstrap complet est tenté. Cela permet de charger un ensemble de programmes de diagnostics plus avancés. La valeur réelle obtenue en lisant l'adresse d'E/S 379h est la valeur en complément à 1 ce qui fait que la fourchette des valeurs (masquées avec 07h) s'étend de 7 (pour l'anglais) à 0.

3.2 UTILISATION DE LA MEMOIRE PAR LE PROCESSEUR

Le tableau suivant reprend la description de l'utilisation de l'espace mémoire sous une forme différente avec les zones occupées par les interruptions et le ROS.

Adresses	Usage
00000 à 003FF	Vecteurs d'interruption 0 à 255 du processeur. Pour trouver l'adresse de départ d'un vecteur particulier, multipliez son numéro par 4.
00400 à 00500	Variables ROS (voir section 2.4)
00501 à 9FFFF	Zone système (ou utilisateur) de la RAM. La zone de 640 K (y compris les deux entrées précédentes de ce tableau) est la quantité normale mémoire installée dans le système.
A0000 à BFFFF	128 K de RAM réservés la vidéo (IGA ou autres cartes). Voir les tampons en 1.11.
C0000 à C3FFF	Zone de la ROM BIOS IGA. Les 2000h premiers octets comprennent du code exécutable et les autres les tables de référence des modes et les jeux de caractères. Cette zone de ROM (et la suivante en C4000-C7FFF) peut être inhibée en positionnant le sw10 sur ON.

Adresses	Usage
C4000 à C7FFF	Zone des caractères étrangers en ROM. On y trouve les fontes danoises, portugaises et greques. Cette zone peut être inhibée en positionnant le sw9 sur OFF.
C8000 à C9FFF	Zone de la ROM BIOS de base du contrôleur de disque dur.
CA000 à EFFFF	Cette zone est utilisée par les contrôleurs de disques durs externes et les cartes périphériques (comme les LAN) ayant besoin d'une ROM BIOS. Quand elle est installée, la ROM de diagnostic Amstrad réside en E0000-E7FFF.
F0000 à FFFFF	Zone de 64 K réservée à la ROM système. Le ROS réside dans les 16 K de FC000 à FFFFF. Les 48 K restant étant réservés à de futures extensions. Les 16 K de la ROM ROS sont répétés 4 fois dans cette zone.

3.3 TOUCHES ET CODES DE TOUCHES

Le tableau suivant correspond à un clavier américain.

Code touche	Hex	Touche Maj. (US)
1	01	ESC
2	02	1 et !
3	03	2 et @
4	04	3 et #
5	05	4 et $
6	06	5 et %

Code touche	Hex	Touche Maj. (US)
49	31	N
50	32	M
51	33	, et <
52	34	. et >
53	35	/ et ?
54	36	SHIFT droit

Code touche	Hex	Touche Maj. (US)
7	07	6 et 4
8	08	7 et &
9	09	8 et *
10	0A	9 et (
11	0B	0 et)
12	0C	- et _
13	0D	= et +
14	0E	<-
15	0F	TAB
16	10	Q
17	11	W
18	12	E
19	13	R
20	14	T
21	15	Y
22	16	U
23	17	I
24	18	O
25	19	P
26	1A	[et {
27	1B	] et }
28	1C	CR
29	1D	CTRL
30	1E	A
31	1F	S
32	20	D
33	21	F
34	22	G
35	23	H
36	24	J
37	25	K
38	26	L
39	27	; et :
40	28	' et "
41	29	# et ~

Code touche	Hex	Touche Maj. (US)
55	37	PRTSC
56	38	ALT
57	30	ESPACE
58	38	CAPS LOCK
59	3B	F1
60	3C	F2
61	3D	F3
62	3E	F4
63	3F	F5
64	40	F6
65	41	F7
66	42	F8
67	43	F9
68	44	F10
69	45	Num Lock
70	46	Scroll Lock
71	47	7 Numérique
72	48	8 Numérique
73	49	9 Numérique
74	4A	- Numérique
75	4B	4 Numérique
76	4C	5 Numérique
77	4D	6 Numérique
78	4E	+ Numérique
79	4F	1 Numérique
80	50	2 Numérique
81	51	3 Numérique
82	52	0 Numérique
83	53	. Numérique
84-111	54-6F	indéfini
112	70	DEL ->
113-115	71-73	indéfini
116	74	ENTER
117-118	75-76	indéfini
119	77	JOY. FEU2

Code touche	Hex	Touche Maj. (US)
42	2A	SHIFT gauche
43	2B	\ et |
44	2C	Z
45	2D	X
46	2E	C
47	2F	V
48	30	B

Code touche	Hex	Touche Maj. (US)
120	78	JOY. Feu1
121	79	JOY. Droite
122	7A	JOY. Gauche
123	7B	JOY. Bas
124	7C	JOY. Haut
125	7D	Souris M2
126	7E	Souris M1
127	7F	indéfini

3.4 REGISTRE DU PROCESSEUR DE COMMUNICATIONS ASYNCHRONES (8250)

Pour un travail important à ce sujet, nous recommandons que le concepteur se procure la documentation du INS 8250. Les tableaux suivants présentent les registres accessibles principaux. Aucune dénomination officielle en français n'étant établie, nous avons conservé les noms américains. Le lecteur pourra se reporter à l'annexe 4 pour y trouver la signification des différents signaux qui n'est pas toujours en rapport direct avec le nom qu'ils portent.

Registre Modem Status (MSR) [R6]. Adresse d'E/S 3FEh

Bit(s)	Fonction
7	Data Carrier Detect (DCD)
6	Ring Indicator (RI)
5	Data Set Ready (DSR)
4	Clear To Send (CTS)
3	Delta Data Carrier Detect (DDCD)
2	Trailling Edge Ring Indicator (TERI)
1	Delta Data Set Ready (DDSR)
0	Delta Clear To Send (DCTS)

Registre Line Status (LSR) [R5]. Adresse d'E/S 3FDh

Bit(s)	Fonction
7	Toujours 0
6	Transmitter Shift Register Empty (TSRE)
5	Transmitter Holding Register Empty (THRE)
4	Break Interrupt (BI)
3	Framing Error (FE)
2	Parity Error (PE)
1	Overrun Error (OE)
0	Data Ready (DR)

Registre Modem Control (MCR) [R4]. Adresse d'E/S 3FCh

Bit(s)	Fonction
7	Toujours 0
6	Toujours 0
5	Toujours 0
4	Loop (mode diagnostic)
3	Out 2 (bouclée sur RI)
2	Out 1 (bouclée sur DCD)
1	Request To Send (RTS demande d'émission) (bouclée sur DSR)
0	Data Terminal Ready (DTR) (bouclée sur CTS)

Registre Line Control (LCR) [R3]. Adresse d'E/S 3FBh

Bit(s)	Fonction
7	Divisor Latch Access (DLAB) (sélectionne registres 0 et 1)
6	Etablissement du break (Set Break)
5	Stick Parity (Maintient la parité puisque EPS ne le fait pas si PEN = 1)
4	Even Parity Select (EPS sélection parité paire)
3	Parity Enable (PEN activation parité)
2	Nbre de bits d'arrêt (STB ; 0=1 bit d'arrêt, 1=>1)
1	Bit n°1 de sélection de la longueur du mot (WLS1). (Bits 0 à 3 = Bits 5 à 8)
0	Bit n°0 de sélection de la longueur du mot (WLS0)

Registre Interrupt Identification (IIR) [R2]. Adresse d'E/S 3FAh

Bit(s)	Fonction
7	Toujours 0
6	Toujours 0 Type d'interruption IID
5	Toujours 0 3Rx Line Status
4	Toujours 0 2 Rx. Data Avail.
3	Toujours 0 1 Reg. TxH. Vide
2	Bit 1 d'indentification d'interruption (IID1)
1	Bit 0 d'indentification d'interruption (IID0)
0	Pas d'interruption en attente

Registre Interrupt Enable (IER) [DLAB=0 : R1]. Adresse d'E/S 3F9h

Lorsque le Divisor Access Latch Bit (registre Line Control, bit 7 : DLAB) est à 0, lire l'adresse d'E/S 3F9h donne le IER.

Bit(s)	Fonction
7	Toujours 0
6	Toujours 0
5	Toujours 0
4	Toujours 0
3	Etat Modem (EDSSI)
2	Etat ligne du receveur (ELSI)
1	Registre Transmitter Holding vide (ETBEI)
0	Données reçues disponibles (ERBAI)

Registre Receive Buffer (RBR) et registre Transmit Holding (THR) [DLAB=0 : R0]. Adresse d'E/S 3F8h

Lorsque le Divisor Access Latch Bit (registre Line Control, bit 7 : DLAB) est à 0, lire et écrire à l'adresse d'E/S 3F8 donne accès aux registres RBR et THR. Une lecture de 3F8 donne le RBR (bits 0 à 7). Une écriture sur 3F8 écrit dans le registre THR.

Divisor Latches MS et LS (DLL et DLM) [R0 et R1 quand DLAB = 1]

Lorsque le Divisor Access Latch Bit (registre Line Control, bit 7 : DLAB) est à 1, les registres 0 et 1 sont les registres 16 bits Divisor. Les bits de poids faible sont positionnés en écrivant à l'adresse d'E/S 3F8 et les bits de poids fort en écrivant à l'adresse 3F9. Les diviseurs et leurs vitesses de transmission respectives sont les suivantes :

Vitesse (baud)	Diviseur	R1 et R0 (hexa)
75	1536	06-00
300	384	01-80
600	192	00-C0
1200	96	00-60
2400	48	00-30
4800	24	00-18
9600	12	00-0C

3.5 REGISTRE DU CONTROLEUR DMA HAUTES PERFORMANCES (8237A-4)

Voici les principaux registres accessibles du 8237A. Nous recommandons au concepteur d'un travail important de se procurer la documentation de ce processeur.

Registre Command - Adresse d'E/S en écriture 008h

Bit(s)	Fonction (Action ...{1/0})
7	DACK (signal d'accord d'accès direct) actif {haut/bas}
6	DREQ (signal de demande d'accès direct) actif {haut/bas}
5	Sélection d'écriture {étendue/la dernière}
4	Priorité {tournante/fixe}
3	Timing {comprimé (plus rapide)/normal}
2	Contrôleur {inhibé/autorisé}
1	maintient adresse voie 0 {autorisé/inhibé}
0	Mémoire à mémoire IMPOSSIBLE SUR PC 1640 {autorisé/inhibé}

Registre Status - Adresse d'E/S en lecture 008h

Bit(s)	Fonction
7	Requète canal 3
6	Requète canal 2
5	Requète canal 1
4	Requète canal 0
3	Décompte achevé sur canal 3
2	Décompte achevé sur canal 2
1	Décompte achevé sur canal 1
0	Décompte achevé sur canal 0

Registre Mode - Adresse d'E/S 00Bh [Ecriture seule]

Bit(s)	Fonction
7	bit 1 de sélection de mode (modes 0 = demande, 1 = unique) (modes 2 = bloc, 3 = cascade)
6	bit 0 de sélection de mode
5	Adresse {décrémentée/incrémentée}
4	Autoinitialisation {autorisée/inhibée}
3	bit 1 du type de transfert (types 0 = vérification, 1 = écriture) (types 2 = lecture, 3 Interdit)
2	bit 0 du type de transfert
1	bit 1 de sélection de canal (canaux de 0 à 3 respectivement)
0	bit 0 de sélection de canal

Registre Request - Adresse d'E/S 009h [Ecriture seule]

Bit(s)	Fonction
7	Indifférent
6	Indifférent
5	Indifférent
4	Indifférent
3	Indifférent
2	Bit de requète {VRAI/FAUX}
1	bit 1 de sélection de canal
	(canaux de 0 à 3 respectivement)
0	bit 0 de sélection de canal

Registre Mask Set/Reset - Adresse d'E/S 00Ah [Ecriture seule]

Bit(s)	Fonction
7	Indifférent
6	Indifférent
5	Indifférent
4	Indifférent
3	Indifférent
2	bit de masque {VRAI/FAUX}
1	bit 1 de sélection de canal
	(canaux de 0 à 3 respectivement)
0	bit 0 de sélection de canal

Registre Mask Write - Adresse d'E/S 00Fh [Ecriture seule]

Bit(s)	Fonction
7	Indifférent
6	Indifférent
5	Indifférent
4	Indifférent
3	Bit de masque du canal 3 {1/0}
2	Bit de masque du canal 2 {1/0}
1	Bit de masque du canal 1 {1/0}
0	Bit de masque du canal 0 {1/0}

3.6 MOTS D'INITIALISATION DU CONTROLEUR D'INTERRUPTIONS PROGRAMMABLE (8259A-2)

La séquence d'initialisation (ICW pour Initialisation Command Word) se compose des 4 mots suivants :

Mot d'initialisation ICW1 - Adresse d'E/S en écriture 020h

Bit(s)	Fonction (Action ...{1/0})
7	Non disponible
6	Non disponible
5	Non disponible
4	Toujours 1
3	Mode de déclenchement {niveau/front}
2	Intervalle d'adresse d'appel {4/8}
1	Mode {solo/cascadé} (ICW3 nécessaire en solo)
0	ICW4 {nécessaire/inutile}

Mot d'initialisation ICW2 - Adresse d'E/S en écriture 021h

Bit(s)	Fonction
7	Bit 7 du niveau d'interruption (T7) voir remarques
6	Bit 6 du niveau d'interruption (T6)
5	Bit 5 du niveau d'interruption (T5)
4	Bit 5 du niveau d'interruption (T4)
3	Bit 5 du niveau d'interruption (T3)
2-0	Inutilisés

Cet octet définit l'une des adresses de vecteurs d'interruption (dans la zone des adresses absolues 0 à 3FFh) comme devant être utilisée lors de l'interruption provoquée. Les bits Type 3 à 7 (maintenus sur le bus de données durant le cycle INTA) représentent les bits 5 à 9 de l'adresse du vecteur d'interruption choisi. Les 3 bits de poids faible sont positionnés par le 8259A en fonction du niveau d'interruption.

Mot d'initialisation ICW3 - Adresse d'E/S en écriture 021h

Ce mot d'initialisation n'est pas utilisé étant donné que Single (bit 1 de ICW1) est toujours vrai avec un PC 1640. En d'autres circonstances, ce mot d'initialisation détermine quel IR aura un esclave en mode Master et pour l'esclave, les bits 0 à 3 donnent le numéro d'identification (0 à 7).

Mot d'initialisation ICW4 - Adresse d'E/S en écriture 021h

Bit(s)	Fonction
7	Toujours 0
6	Toujours 0
5	Toujours 0
4	Mode spécial "Fully Nested" {autorisé/inhibé}
3	Mode bufférisé {ON/OFF}
2	Mode {Maître/Esclave} (n'est valide qu'en mode bufférisé)
1	EOI {Automatique/normale}
0	Toujours 1 puisqu'en mode 8086/8088

Les mots de commande d'opérations

Ces mots de commande (OCW pour Operation Command Words) permettent de sélectionner différents modes de fonctionnement du 8259A.

Mot de commande d'opérations OCW1 - Adresse d'E/S en écriture 021h

Bit(s)	Fonction (Action ...{1/0})
7	Masque d'interruption 7 {1/0}
6	Masque d'interruption 6 {1/0}
5	Masque d'interruption 5 {1/0}
4	Masque d'interruption 4 {1/0}
3	Masque d'interruption 3 {1/0}
2	Masque d'interruption 2 {1/0}
1	Masque d'interruption 1 {1/0}
0	Masque d'interruption 0 {1/0}

Ces 8 bits (masques) peuvent soit masquer (c'est-à-dire inhiber quand M = 1), soit autoriser leurs canaux respectifs.

Mot de commande d'opérations OCW2 - Adresse d'E/S en écriture 020h

Bit(s)	Fonction
7	Bit de rotation (R)
6	Bit de spécification (SL)
5	Bit de fin d'interruption (EOI)
4	Toujours 0
3	Toujours 0
2	Bit 2 du niveau (L2)
1	Bit 1 du niveau (L1)
0	Bit 0 du niveau (L0)

Les level bits (bits du niveau) sont nécessaires quand Specific (SL) est mis à 1.

Mot de commande d'opérations OCW3 - Adresse d'E/S en écriture 020h

Bit(s)	Fonction (Action ...{1/0})
7	Toujours 0
6	Active le mode spécial de masque (ESMM)
5	Mode spécial de masque (SMM) {établi/reset}
4	Toujours 0
3	Toujours 1
2	Etablit le mode interruptions par balayage (Polling) {autorisé/inhibé}
1	Lecture du registre {IS/IR} au prochain signal -RD (RIS)

Le bit ESMM doit être mis à 1 pour que le bit SMM agisse. De la même manière, le bit RR doit être à 1 pour que le bit RIS agisse.

3.7 LES REGISTRES DU TIMER PROGRAMMABLE (8253)

Le PIT (Programmable Interval Timer) 8253 présente quatre éléments accessibles : les 3 compteurs (0 à 2) où les opérations de lecture/écriture se font sur 8 bits (adresses d'E/S 40h-42h) et le registre de mots de commandes (Control Word Register) vers lequel on ne peut qu'écrire (par 043h).

Constitution du registre de mots de commandes (Control Word) :

Bit(s)	Fonction (Action ...{1/0})
7	Bit 1 de sélection de compteur (SC1)
6	Bit 0 de sélection de compteur (SC0)
5	Bit 1 de lecture/chargement (RL1)
4	Bit 0 de lecture/chargement (RL0)
3	Bit 2 du mode (M2)
2	Bit 1 du mode (M1)
1	Bit 0 du mode (M0)
0	{active/désactive} le compteur décimal codé binaire

Les bits SC sélectionnent les compteurs 0 à 2 (la valeur 3 est illicite).

Les bits RL permettent les opérations de lecture/écriture sur les compteurs de la manière suivante :

0 : Verrouillage du compteur (Counter Latching). Permet de saisir au vol (c'est-à-dire de copier vers un registre annexe) la valeur courante d'un compteur.
1 : Lit/Ecrit (R/L) l'octet de poids fort seulement.
2 : Lit/Ecrit l'octet de poids faible seulement.
3 : Lit/Ecrit l'octet de poids faible puis l'octet de poids fort.

Les bits Mode permettent de sélectionner l'un des cinq modes de fonctionnement effectivement disponibles (les modes 6 et 7 reviennent à utiliser les modes 2 et 3).

0 : Interrupt on Terminal count. Envoi d'une demande d'interruption en fin de décomptage.
1 : Programmable One-Shot. Programmable à une passe.
2 : Rate Generator. Générateur de fréquences.
3 : Square Wave generator. Générateur de signaux carrés.
4 : Software Triggered Strobe. Déclenchement logiciel cadencé.
5 : Hardware Triggered Strobe. Déclenchement matériel cadencé.

3.8 REGISTRES DE L'HORLOGE TEMPS REEL RTC HD146818

Le HD146818 est un périphérique CMOS qui présente trois caractéristiques essentielles : une gestion de l'heure complète avec alarme et calendrier sur un siècle, une interruption périodique programmable avec un générateur d'onde carrée et enfin une RAM statique à faible consommation de 50 octets.

La figure ci-dessous est la carte mémoire du HD146818. On trouve 50 octets de RAM, 10 octets de RAM qui normalement reçoivent l'heure, la date, l'heure d'alarme avec quatre octets d'état et de contrôle. Tous ces octets sont directement accessibles en lecture et écriture au processeur à l'exception des registres C et D qui ne fonctionnent qu'en lecture seule. Le bit 7 du registre A et l'octet des secondes sont eux aussi RO (Read Only).

0	Secondes	00
1	Seconde de l'alarme	01
2	Minutes	02
3	Minute de l'alarme	03
4	Heures	04
5	Heure d'alarme	05
6	Jour de la semaine	06
7	Quantième	08
8	Mois	08
9	Année	09
10	Registre A	0A
11	Registre B	0B
12	Registre C	0C
13	Registre D	0D

14		0E
	50 octets de RAM utilisateur	
63		3F

3.8.1 Adresses de l'heure, de la date et de l'alarme

Le processeur obtient l'heure et les informations complètes sur la date en lisant les adresses appropriées. Le programme peut modifier l'heure, la date et l'alarme en écrivant dans ces adresses. Le contenu des 10 octets de l'heure, de la date et de l'alarme peut soit être binaire, soit décimal codé binaire (BCD).

Avant que l'on puisse initialiser les registres internes, le bit SET du registre B doit être porté à 1 pour éviter qu'une mise à jour de ces éléments n'intervienne. Le programme initialise les 10 adresse dans le format binaire choisi (BCD ou pas) puis indique ce format grâce au bit Data Mode (DM) du registre B. Toutes ces 10 adresses doivent avoir ce même format. On peut alors remettre à 0 le bit SET pour permettre une mise à jour de l'heure et de la date. La manière dont celle-ci se passe dépend du mode de données choisi. Ce mode de données ne peut être changé sans une nouvelle initialisation des 10 octets de données.

Le tableau de la page suivante présente les formats binaire et BCD des adresses d'heure, de date et d'alarme.

Numéro d'octet	Fonction	Four-chette	Mode binaire	Mode BCD
0	Secondes	0-59d	00-3B	00-59
1	Sec. Alarme	0-59d	00-3B	00-59
2	Minutes	0-59d	00-3B	00-59
3	Min. Alarme	0-59d	00-3B	00-59
4	Heures Mode 12 H.	1-12d	01-0C (AM) 81-8C (PM)	01-12 (AM) 81-92(PM)
	Mode 24 H.	0-23d	00-17	00-23
5	H. Alarme Mode 12 H.	1-12d	01-0C (AM) 81-8C (PM)	01-12 (AM) 81-92(PM)
	Mode 24 H.	0-23d	00-17	00-23
6	Jour semaine	1-7d	01-07	01-07
7	Quantième	1-31d	01-1F	01-31
8	Mois	1-12d	01-0C	01-12
9	Année	0-99d	00-63	00-99

Toutes les valeurs sauf celles suivies d'un d minuscule sont hexadécimales. Pour le jour de la semaine, Dimanche = 1.

Le bit de mode 24/12 du registre B établit que le comptage des heures se fera dans la fourchette 1 à 12 ou 0 à 23. Ce bit ne peut être modifié sans ré-initialiser l'adresse des heures. En format 12 heures, le bit de poids fort de l'heure indique PM lorsqu'il est à 1. Toutes ces données (heures complète, date complète et alarme) ne sont pas accessibles en permanence au processeur. Une fois par seconde, ces 10 octets passent sous le contrôle du processus de mise à jour qui avance l'horloge d'une seconde et teste s'il doit déclencher l'alarme. Si l'on effectue la lecture de l'une de ces 10 adresses à cet instant, la valeur retournée est indéfinie. Le bit Update In Progress (UIP), mise à jour en cours, permet de savoir si ce processus est en cours ou pas.

Le bit UPI est porté à 1 une fois par seconde et le cycle de mise à jour commence 244 uS plus tard. Par conséquent, si UPI est à 0, l'utilisateur a au moins 244 uS pour lire ses données et avant que les valeurs soient changées.

3.8.2 Les adresses des registres de la RTC

Le HD146818 possède quatre registres accessibles au processeur. Ils sont pleinement accessibles même durant le processus de mise à jour.

L'attribution des bits du registre A (adresse 0Ah) est la suivante :

Bit(s)	Fonction
7	Mise à jour en cours (Update in Progress : UIP)
6	Bit 2 du diviseur (DV2)
5	Bit 1 du diviseur (DV1)
4	Bit 0 du diviseur (DV0)
3	Bit 3 de sélection de fréquence (RS3 : Rate selection)
2	Bit 2 de sélection de fréquence (RS2)
1	Bit 1 de sélection de fréquence (RS1)
0	Bit 0 de sélection de fréquence (RS0)

Le bit UPI permet de savoir si les 10 octets de l'heure, de la date et de l'alarme vont être mis à jour ou pas comme expliqué dans la section précédente.

Les trois Divider Bits (DV2-DV0) permettent d'intentifier la fréquence de base de temps (parmi 3) actuellement utilisée ou de reseter la chaîne de division.

Les 4 bits de sélection de féquence (RS3-RS0) permettent de choisir l'un des 15 tempos (tap) sur les 22 fréquences fournies par la chaîne de division. Ce tap peut servir à générer un signal sur la broche SQW et/ou une interruption périodique.

Constitution du registre N (adresse 0Bh) :

Bit(s)	Fonction
7	Bit SET
6	Bit d'autorisation d'interruption périodique (PIE)
5	Bit d'activation de l'interruption d'alarme (AIE)
4	Bit d'activation de l'interruption de fin de mise à jour (UIE)
3	Bit d'activation de l'onde carrée (SQWE)
2	Mode des données (DM)
1	Format 24/12 heures
0	Activation heure d'hiver/d'été (DSE)

Quand le bit SET est à 0, le cycle de mise à jour fait normalement avancer le comptage une fois par seconde. Lorsque l'on affecte la valeur 1 à ce bit SET, le cycle de mise à jour est abandonné et le processeur peut initialiser les adresses d'heure et de date sans risquer la perturbation provoquée par la mise à jour. SET est un bit accessible en lecture/écriture et il n'est pas modifié par -RES ou par une fonction interne du HD146818.

Le bit PIE est accessible en lecture/écriture et il autorise le bit indicateur d'interruption périodique (PF) à faire en sorte que la broche -IRQ soit forcée au niveau bas. Le programme peut écrire un 1 dans PIE afin de recevoir une interruption périodique au rythme spécifié par les bits RS3-RS0 du registre A. Un 0 dans PIE interdit la génération de 'IRQ mais le bit indicateur PF passe toujours à 1 à la fréquence définie.

Le bit AIE est accessible en lecture/écriture. Lorsqu'il est à 1, il autorise le flag d'alarme (AF) à forcer -IRQ. Une interruption d'alarme intervient à chaque seconde où les trois octets de temps (heure, minutes, secondes) sont égaux aux trois octets d'alarme. Quand AIE est à 0, le bit AF ne génère pas -IRQ. La broche -RES met à zéro AIE. Les fonctions internes n'affectent pas le bit AIE.

Le bit UIE (en lecture/écriture) autorise le bit de fin de mise à jour (UF pour Update end Flag) à provoquer -IRQ. UIE passe à 0 si la broche -RES passe au niveau bas ou si le bit SET passe à 1.

Lorsque le bit SQWE (accès en lecture/écriture) est mis à 1 par le processeur, un signal carré dont la fréquence est spécifiée par la configuration des bits RS3 à RS0 est émis par la broche SQW. Quand SQWE est mis à 0, la broche est maintenue au niveau bas. Le bit SQWE est effacé par la broche -RES.

Le bit DM détermine si les données de date et d'heure seront exprimées en format binaire ou décimal codé binaire. DM à 1 correspond aux données binaires et DM à 0 au DCB. DM est un bit en lecture/écriture qui n'est pas modifié par -RES ou les fonctions internes du HD146818.

Le bit 24/12 spécifie le format des octets d'heure. Un 1 indique le mode 24 heures et un 0 le mode 12 heures. C'est un bit accessible en lecture/écriture qui n'est pas affecté par -RES ou les fonctions internes du HD146818.

Le bit DSE (lecture/écriture), lorsqu'il est à un met en fonction le mode heure d'été/heure d'hiver (daylight mode). Dans ce mode, l'heure passe directement de 1:59:59 à 3:00:00 AM le dernier dimanche d'Avril. Le dernier dimanche d'Octobre l'heure est décrémentée : elle passe de 1:59:59 à 1:00:00. DSE n'est pas modifié par -RES ou les fonctions internes du HD146818.

Les bits du registres C (adresse 0Ch) sont classifiés comme suit :

Bit(s)	Fonction
7	Flag de requète d'interruption (IRQF)
6	Flag d'interruption périodique (PF)
5	Flag d'interruption d'alarme (AF)
4	Flag d'interruption de fin de mise à jour (UF)
3-0	0

Le registre C est en lecture seule et n'est donc pas affecté par une opération d'écriture tentée par un programme.

La valeur du bit IRQF est issue de l'équation logique : IRQF = PF.PIE + AF.AIE + UF.UIE. A chaque fois que le bit IRQF est à 1, la broche -IRQ est amenée au niveau bas. Tous les flags du registre C sont effacés après une lecture par le programme ou lorsque -RES est au niveau bas.

Le bit PF est mis à 1 lorsqu'un front particulier du tap sélectionné (par RS3 à RS0) est détecté dans la chaîne de division. Le bit PF est mis à 1 de façon indépendante de l'état du bit PIE.

Le bit AF est mis à 1 lorsque l'heure courante devient égale à l'heure d'alarme.

Le bit UF est mis à 1 après chaque cycle de mise à jour.

Les bits restant (3 à 0) sont toujours à 0.

Le registre D est constitué ainsi :

Bit(s)	Fonction
7	Valid RAM Time (heure RAM valide) : VRT
6 à 0	0

Le bit VRT indique que l'heure et le contenu de la RAM sont valides. Le bit VTR est mis à 0 lorsque la broche PS est au niveau bas. Le microprocesseur peut forcer VRT à 1 après une initialisation des paramètres d'heure et de date pour indiquer qu'ils sont valides. Le VRT est accesible en lecture seule et n'est pas affecté par l'état de la broche -RES. Pour le mettre à 1, il faut aller lire le registre D.

Les bits 0 à 6 sont inutilisés et toujours à 0.

3.9 LE CONTROLEUR DE LECTEURS DE DISQUETTES uPD765A

Le contrôleur de lecteurs de disquettes (FDC pour Floppy Disk Controler) possède deux registres qui sont accessibles au CPU. Le Main Status Register (Registre d'état) à l'adresse 03F4h et le Data Register (registre de données) en 3F5h. Tous deux ont une longueur de 8 bits. Le Status Register indique l'état du FDC et on peut y accéder à tout moment. En réalité, le Data Register est constitué de plusieurs registres assemblés en une pile et reçoit des données, les commandes et la valeur d'état du Floppy Disk Drive (FDD). Pour programmer une commande particulière, on écrit une valeur dans le registre de données. Symétriquement, on lit l'adresse de données pour obtenir le résultat d'un opération. Le registre d'état ne peut qu'être lu et sert à faciliter les transferts entre le CPU et le contrôleur de lecteurs de disquettes.

Le uPD765A peut effectuer 15 commandes différentes. Chacune d'elle est spécifiée sur plusieurs octets. Le résultat d'une opération peut lui aussi être relu par le processeur en plusieurs octets. A cause de cet aspect "multi-octets" des échanges entre le processeur et le FDC, il est plus commode de considérer que l'envoi d'un ordre se décompose en trois phases :

La phase de commande : Le processeur envoie toutes les informations nécessaires au FDC pour qu'il effectue le travail demandé.

La phase d'exécution : Le FDC effectue l'opération.

La phase de résultat : Une fois l'opération réalisée, le CPU dispose des états et des informations d'intendance.

Le uPD765A possède cinq registres d'état : le registre d'état principal, déjà mentionné, qui peut donc être lu à tout moment et quatre registres d'état (ST0, ST1,ST2, ST3) correspondant à une phase donnée du travail du FDC et qui ne sont disponibles que dans la phase de résultat après l'exécution d'une commande.

C'est l'opération demandée qui détermine quels seront les registres d'état retournés.

Les octets de commande qui sont envoyés au uPD765A durant la phase de commande doivent l'être dans l'ordre donné par le tableau de la commande présenté ici. C'est-à-dire que le code de commande doit être envoyé en premier, suivi des autres octets selon la séquence définie. Aucun raccourci n'est autorisé que ce soit en phase de commande ou de résultats. Après l'envoi du dernier octet de commande, la phase d'exécution démarre automatiquement. Dans le même esprit, dès que le dernier octet retourné est lu durant la phase de résultats, la commande est automatiquement achevée et le uPD765A est prêt pour une autre commande.

Il faut bien noter que durant la phase de résultats, tous les octets figurant dans la description de la commande doivent être lus. Par exemple, la commande Read Data retourne 7 octets durant la phase de résultats : il faut les lire tous avant de pouvoir lancer une nouvelle commande.

Les registres d'état :

Main Status Register

Bit(s)	Fonction
7	Request for master (RQM) requète au maître
6	Données E/S (DIO)
5	Mode d'exécution (EXM)
4	FDC occupé (CB)
3	Lecteur 3 occupé (D3B)
2	Lecteur 2 occupé (D2B)
1	Lecteur 1 occupé (D1B)
0	Lecteur 0 occupé (D0B)

Status Register 0 (ST0)

Bit(s)	Fonction
7	Bit 1 du code d'interruption (IC1)
6	Bit 0 du code d'interruption (IC0)
5	Fin de recherche (seek end) (SE)
4	Vérification de l'équipement (EC)
3	Non prêt (NR)
2	Désignation tête (HD)
1	Sélection de l'unité 1 (US1)
0	Sélection de l'unité 2 (US2)

Status Register 1 (ST1)

Bit(s)	Fonction
7	Fin de cylindre (EN)
6	Toujours 0
5	Data error (DE)
4	Overrun (OR)
3	Toujours 0
2	No Data (ND)
1	Non writable (écriture impossible) (NW)
0	Address Mark manquante (MA)

Status Register 2 (ST2)

Bit(s)	Fonction
7	Toujours 0
6	Control Mark (CM)
5	Erreur de donnée dans champ de données (DD)
4	Erreur de désignation de cylindre (WC)
3	Scan Equal Hit (SH)
2	Scan Not Satisfied (SN)
1	Mauvais cylindre (BC)
0	Absence de l'Address Mark dans champ de données (MD)

Status Register 3 (ST3)

Bit(s)	Fonction
7	Erreur (FT)
6	Protection en écriture (WP)
5	Prêt (RY)
4	Piste 0 (T0)
3	Désignation de la tête (HD)
1	Sélection de l'unité 1 (US1)
0	Sélection de l'unité 0 (US0)

Liste des commandes :

Read Data (lecture de données)

Phase de commande : 9 octets

Octet 1 : Code de commande

Bit(s)	Fonction (Action ...{1/10})
7	(MT) Multi pistes {active/inhibe}
6	(FM) Sélection {MFM/FM} (simple/double densité)
5	(SK) Active le mode saut des adresses de données effacées
4	0
3	0
2	1
1	1
0	0

Octet 2 : Sélection de la tête et de l'unité

Bit(s)	Fonction
7-3	Indifférent
2	HD Sélecteur de tête (0 ou 1)
1	US1
0	US0 Sélecteur d'unité (0 ou 1)

Octet 3 : Numéro de cylindre (0-76)
Octet 4 : Numéro de tête (comme indiqué dans le champ ID)
Octet 5 : Secteur à lire
Octet 6 : Nombre d'octets par secteur
Octet 7 : EOT (End Of Track) Numéro du denier secteur de piste
Octet 8 : GPL (Gap3 Length) Longueur du gap3
Octet 9 : DTL Longueur des données à lire

Durant l'exécution, les données sont transférées entre le FDD (Floppy Disk Drive) et la mémoire du CPU.

La phase de résultats retourne 7 octets :

Octet 1 : ST0 (Voir la table du Status Register 0)
Octet 2 : ST1 (Voir la table du Status Register 1)

Octet 3 : ST2 (Voir la table du Status Register 2)
Octet 4 : Numéro de cylindre final
Octet 5 : Numéro de la tête à la dernière lecture
Octet 6 : Dernier secteur lu
Octet 7 : Nombre d'octets lus

Read Track (lecture de piste)

Phase de commande : 9 octets

Octet 1 : Code de commande

Bit(s)	Fonction (Action ...{1/0})
7	0
6	(FM) Sélection {MFM/FM} (simple/double densité)
5	(SK) Active le mode saut des adresses de données effacées
4	0
3	0
2	0
1	1
0	0

Octet 2 : Sélection de la tête et de l'unité

Bit(s)	Fonction
7-3	Indifférent
2	HD Sélecteur de tête (0 ou 1)
1	US1
0	US0 Sélecteur d'unité (0 ou 1)

Octet 3 : Numéro de cylindre (0-76)
Octet 4 : Numéro de tête (comme indiqué dans le champ ID)

Octet 5 : Secteur à lire
Octet 6 : Nombre d'octets par secteur
Octet 7 : EOT (End Of Track) Dernier numéro de secteur de piste
Octet 8 : GPL (Gap3 Length) Longueur du gap3
Octet 9 : DTL Longueur des données à lire

Durant l'exécution, les données sont transférées entre le FDD (Floppy Disk Drive) et la mémoire du CPU. Le contrôleur lit tous les champs de données depuis le trou d'index jusqu'à EOT.

La phase de résultats retourne 7 octets :

Octet 1 : ST0 (Voir la table du Status Register 0)
Octet 2 : ST1 (Voir la table du Status Register 1)
Octet 3 : ST2 (Voir la table du Status Register 2)
Octet 4 : Numéro de cylindre final
Octet 5 : Numéro de la tête de la dernière lecture
Octet 6 : Dernier secteur lu
Octet 7 : Nombre d'octets lus

Read deleted data (lecture de données effacées)

Phase de commande : 9 octets

Octet 1 : Code de commande

Bit(s)	Fonction (Action ...{1/10})
7	(MT) Multi pistes {active/inhibe}
6	(FM) Sélection {MFM/FM} (simple/double densité)
5	(SK) Active le mode saut des adresses de données effacées
4	0
3	1
2	1
1	0
0	0

Octet 2 : Sélection de la tête et de l'unité

Bit(s)	Fonction
7-3	Indifférent
2	HD Sélecteur de tête (0 ou 1)
1	US1
0	US0 Sélecteur d'unité (0 ou 1)

Octet 3 : Numéro de cylindre (0-76)
Octet 4 : Numéro de tête (comme indiqué dans le champ ID)
Octet 5 : Secteur à lire
Octet 6 : Nombre d'octets par secteur
Octet 7 : EOT (End Of Track) Numéro du dernier secteur de piste
Octet 8 : GPL (Gap3 Length) Longueur du gap3
Octet 9 : DTL Longueur des données à lire

Durant l'exécution, les données sont transférées entre le FDD (Floppy Disk Drive) et la mémoire du CPU.

La phase de résultats retourne 7 octets :

Octet 1 : ST0 (Voir la table du Status Register 0)
Octet 2 : ST1 (Voir la table du Status Register 1)
Octet 3 : ST2 (Voir la table du Status Register 2)
Octet 4 : Numéro de cylindre final
Octet 5 : Numéro de la tête de la dernière lecture
Octet 6 : Dernier secteur lu
Octet 7 : Nombre d'octets lus

Read ID (Lecture d'identificateur)

Phase de commande : 2 octets

Octet 1 : Code de commande

Bit(s)	Fonction (Action ...{1/10})
7	0
6	(FM) Sélection {MFM/FM} (simple/double densité)
5	0
4	0
3	1
2	0
1	1
0	0

Octet 2 : Sélection de la tête et de l'unité

Bit(s)	Fonction
7-3	Indifférent
2	HD Sélecteur de tête (0 ou 1)
1	US1
0	US0 Sélecteur d'unité (0 ou 1)

Durant l'exécution, le premier identificateur correct du cylindre est copié dans le Data Register.

La phase de résultats retourne 7 octets :

Octet 1 : ST0 (Voir la table du Status Register. 0)
Octet 2 : ST1 (Voir la table du Status Register 1)
Octet 3 : ST2 (Voir la table du Status Register 2)
Octet 4 : Cylindre
Octet 5 : Tête
Octet 6 : Secteur
Octet 7 : Nombre d'octets par secteur

Write Data (Ecriture de données)

Phase de commande : 2 octets

Octet 1 : Code de commande

Bit(s)	Fonction (Action ...{1/10})
7	(MT) Multi pistes {active/inhibe}
6	(FM) Sélection {MFM/FM} (simple/double densité)
5	0
4	0
3	0
2	1
1	0
0	1

Octet 2 : Sélection de la tête et de l'unité

Bit(s)	Fonction
7-3	Indifférent
2	HD Sélecteur de tête (0 ou 1)
1	US1
0	US0 Sélecteur d'unité (0 ou 1)

Durant l'exécution, les données sont copiées de la mémoire du CPU vers le FDD (Floppy Disk Drive).

La phase de résultats retourne 7 octets :

Octet 1 : ST0 (Voir la table du Status Register 0)
Octet 2 : ST1 (Voir la table du Status Register 1)
Octet 3 : ST2 (Voir la table du Status Register 2)
Octet 4 : Numéro de cylindre final
Octet 5 : Numéro de la tête de la dernière écriture

Octet 6 : Dernier secteur écrit
Octet 7 : Nombre d'octets écrits

Write Deleted Data (Ecriture de données effacées)

Phase de commande : 9 octets

Octet 1 : Code de commande

Bit(s)	Fonction (Action ...{1/10})
7	(MT) Multi pistes {active/inhibe}
6	(FM) Sélection {MFM/FM} (simple/double densité)
5	0
4	0
3	1
2	0
1	0
0	1

Octet 2 : Sélection de la tête et de l'unité

Bit(s)	Fonction
7-3	Indifférent
2	HD Sélecteur de tête (0 ou 1)
1	US1
0	US0 Sélecteur d'unité (0 ou 1)

Octet 3 : Numéro de cylindre (0-76)
Octet 4 : Numéro de tête (comme indiqué dans le champ ID)
Octet 5 : Secteur
Octet 6 : Nombre d'octets par secteur
Octet 7 : EOT (End Of Track) Dernier numéro de secteur de piste
Octet 8 : GPL (Gap3 Length) Longueur du gap3

Octet 9 : DTL Longueur des données à écrire

Durant l'exécution, les données sont copiées de la mémoire du CPU vers le FDD (Floppy Disk Drive).

La phase de résultats retourne 7 octets :

Octet 1 : ST0 (Voir la table du Status Register 0)
Octet 2 : ST1 (Voir la table du Status Register 1)
Octet 3 : ST2 (Voir la table du Status Register 2)
Octet 4 : Numéro de cylindre final
Octet 5 : Numéro de la tête à la dernière écriture
Octet 6 : Dernier secteur écrit
Octet 7 : Nombre d'octets écrits

Format Track (Formatage d'une piste)

Phase de commande : 6 octets

Octet 1 : Code de commande

Bit(s)	Fonction (Action ...{1/10})
7	0
6	(FM) Sélection {MFM/FM} (simple/double densité)
5	0
4	0
3	1
2	1
1	0
0	1

Octet 2 : Sélection de la tête et de l'unité

Bit(s)	Fonction
7-3	Indifférent
2	HD Sélecteur de tête (0 ou 1)
1	US1
0	US0 Sélecteur d'unité (0 ou 1)

Octet 3 : Nombre d'octets par secteur
Octet 4 : Nombre de secteurs par piste
Octet 8 : GPL Longueur du GAP3 (inter-secteur)
Octet 9 : D Valeur de remplissage

Durant l'exécution, le FDC écrit les en-têtes d'adressage sur toute la piste.

La phase de résultats retourne 7 octets :

Octet 1 : ST0 (Voir la table du Status Register 0)
Octet 2 : ST1 (Voir la table du Status Register 1)
Octet 3 : ST2 (Voir la table du Status Register 2)
Octet 4 : Numéro de cylindre
Octet 5 : Tête
Octet 6 : Secteur
Octet 7 : Nombre d'octets par secteur

Scan equal (Balayage et comparaison d'égalité)

Phase de commande : 9 octets

Octet 1 : Code de commande

Bit(s)	Fonction (Action ...{1/0})
7	(MT) Multi pistes {active/inhibe}
6	(FM) Sélection {MFM/FM} (simple/double densité)
5	(SK) Active le mode saut des adresses de données effacées
4	1
3	0
2	0
1	0
0	1

Octet 2 : Sélection de la tête et de l'unité

Bit(s)	Fonction
7-3	Indifférent
2	HD Sélecteur de tête (0 ou 1)
1	US1
0	US0 Sélecteur d'unité (0 ou 1)

Octet 3 : Numéro de cylindre (0-76)
Octet 4 : Numéro de tête (comme indiqué dans le champ ID)
Octet 5 : Secteur
Octet 6 : Nombre d'octets par secteur
Octet 7 : EOT (End Of Track) Numéro du dernier secteur de piste
Octet 8 : GPL (Gap3 Length) Longueur du gap3
Octet 9 : STP (Step Factor) 1 = Contigus, 2 = Secteurs alternés

Durant l'exécution, les données provenant de la mémoire du CPU sont comparées avec celles du FDD.

La phase de résultats retourne 7 octets :

Octet 1 : ST0 (Voir la table du Status Register 0)
Octet 2 : ST1 (Voir la table du Status Register 1)

Octet 3 : ST2 (Voir la table du Status Register 2)
Octet 4 : Numéro de cylindre final
Octet 5 : Numéro de la tête à la dernière comparaison
Octet 6 : Dernier secteur comparé
Octet 7 : Nombre d'octets comparés.

Scan Low or Equal
(Balayage et comparaison d'infériorité ou d'égalité)

Phase de commande : 9 octets

Octet 1 : Code de commande

Bit(s)	Fonction (Action ...{1/0})
7	(MT) Multi pistes {active/inhibe}
6	(FM) Sélection {MFM/FM} (simple/double densité)
5	(SK) Active le mode saut des adresses de données effacées
4	1
3	1
2	0
1	0
0	1

Octet 2 : Sélection de la tête et de l'unité

Bit(s)	Fonction
7-3	Indifférent
2	HD Sélecteur de tête (0 ou 1)
1	US1
0	US0 Sélecteur d'unité (0 ou 1)

Octet 3 : Numéro de cylindre (0-76)
Octet 4 : Numéro de tête (comme indiqué dans le champ ID)

Octet 5 : Secteur à comparer
Octet 6 : Nombre d'octets par secteur
Octet 7 : EOT (End Of Track) Numéro du dernier secteur de piste
Octet 8 : GPL (Gap3 Length) Longueur du gap3
Octet 9 : STP (Step Factor) 1 = Contigus, 2 = Secteurs alternés

Durant l'exécution, les données provenant de la mémoire du CPU sont comparées avec celles du FDD.

La phase de résultats retourne 7 octets :

Octet 1 : ST0 (Voir la table du Status Register 0)
Octet 2 : ST1 (Voir la table du Status Register 1)
Octet 3 : ST2 (Voir la table du Status Register 2)
Octet 4 : Numéro de cylindre final
Octet 5 : Numéro de la tête de la dernière comparaison
Octet 6 : Dernier secteur comparé
Octet 7 : Nombre d'octets comparés

Scan High or Equal
(Balayage et comparaison de supériorité ou d'égalité)

Phase de commande : 9 octets

Octet 1 : Code de commande

Bit(s)	Fonction (Action ...{1/0})
7	(MT) Multi pistes {active/inhibe}
6	(FM) Sélection {MFM/FM} (simple/double densité)
5	(SK) Active le mode saut des adresses de données effacées
4	1
3	1
2	1
1	0
0	1

Octet 2 : Sélection de la tête et de l'unité

Bit(s)	Fonction
7-3	Indifférent
2	HD Sélecteur de tête (0 ou 1)
1	US1
0	US0 Sélecteur d'unité (0 ou 1)

Octet 3 : Numéro de cylindre (0-76)
Octet 4 : Numéro de tête (comme indiqué dans le champ ID)
Octet 5 : Secteur
Octet 6 : Nombre d'octets par secteur
Octet 7 : EOT (End Of Track) Numéro du dernier secteur de piste
Octet 8 : GPL (Gap3 Length) Longueur du gap3
Octet 9 : STP (Step Factor) 1 = Contigus, 2 = Secteurs alternés

Durant l'exécution, les données provenant de la mémoire du CPU sont comparées avec celles du FDD.

La phase de résultats retourne 7 octets :

Octet 1 : ST0 (Voir la table du Status Register 0)
Octet 2 : ST1 (Voir la table du Status Register 1)
Octet 3 : ST2 (Voir la table du Status Register 2)
Octet 4 : Numéro de cylindre final
Octet 5 : Numéro de la tête de la dernière comparaison
Octet 6 : Dernier secteur comparé
Octet 7 : Nombre d'octets comparés.

Recalibrate (Recalibrage)

Phase de commande : 2 octets

Octet 1 : Code de commande

Bit(s)	Fonction (Action ...{1/0})
7	0
6	0
5	0
4	0
3	0
2	1
1	1
0	1

Octet 2 : Sélection de la tête et de l'unité

Bit(s)	Fonction
7-3	Indifférent
2	0
1	US1
0	US0 Sélecteur d'unité (0 ou 1)

Durant l'exécution, la tête est ramenée sur la piste 0.

La phase de résultats ne retourne pas de valeurs d'état.

Sense Interrupt Status (Lecture des états sur interruption)

Phase de commande : 1 octet

Octet 1 : Code de commande = 08h

La phase de résultats retourne 2 octets :

Octet 1 : ST0 - Status register 0
Octet 2 : PCN - Present Cylinder Number (Position actuelle des têtes)

Specify (Spécification)

Phase de commande : 3 octets

Octet 1 : Code de commande = 03h

Octet 2 : SRT/HUT - Step Rate Time (4 bits de poids fort, par incréments de 1 ms) / Head Unload Time (4 bits de poids faible par incréments de 16 ms).

Octet 3 : HLT/ND - Head Load Time (Bits 1 à 7 par incréments de 2 ms) / Non-DMA Mode (bit 0).

Seek (Positionnement)

Phase de commande : 3 octets

Octet 1 : Code de commande = 0Fh
Octet 2 : Sélection de la tête et de l'unité

Bit(s)	Fonction (Action ...{1/0})
7-3	Indifférent
2	HD Sélecteur de tête (0 ou 1)
1	US1
0	US0 Sélecteur d'unité (0 ou 1)

Octet 3 : Nouveau numéro de cylindre

Durant la phase d'exécution, la tête est positionnée sur le cylindre spécifié.

La phase de résultats ne retourne pas d'états.

Sense Drive Status (Lecture des états)

Phase de commande : 2 octets

Octet 1 : Code de commande = 04h
Octet 2 : Sélection de la tête et de l'unité

Bit(s)	Fonction
7-3	Indifférent
2	HD Sélecteur de tête (0 ou 1)
1	US1
0	US0 Sélecteur d'unité (0 ou 1)

Phase de résultat : 1 octet

Octet 1 : ST3 - Status Rgister 3

Codes opérateurs invalides

Tous les codes de commande ne figurant pas ici sont considérés comme invalides. Lorsque l'on envoie un code de commande invalide, le FDC retourne le registre ST0 avec le bit le plus significatif (Invalid Opcode bit) mis à 1.

LES ANNEXES

ANNEXE 1

INTERFACES LOGICIELLES AVEC LA SOURIS

Le système d'exploitation fourni avec votre PC 1640, MS-DOS 3.2, comprend un ensemble de fonctions permettant à un programme de gérer la souris et de contrôler son curseur. Ces fonctions peuvent être appelées par les applications (grâce aux interfaces logicielles décrites ci-après) à condition que l'on ait exécuté l'utilitaire MOUSE.COM avant le lancement de l'application.

Les fonctions souris détaillées dans cette annexe sont les suivantes :

N°	Fonction
0	Initialisation de la souris
1	Curseur visible
2	Curseur caché
3	Donne la position de la souris et l'état des boutons
4	Fixe la position du curseur souris
5	Donne l'historique des pressions des boutons
6	Donne l'historique des relâchements des boutons
7	Fixe les positions horizontales minimum et maximum du curseur
8	Fixe les positions verticales minimum et maximum du curseur
9	Définition graphique du curseur
10	Définition du curseur texte

N°	Fonction
11	Lecture des compteurs de déplacement de la souris
12	Définition du masque d'appel de la routine utilisateur
13	Activation du mode émulation de crayon optique
14	Désactivation du mode émulation de crayon optique
15	Définition du rapport impulsions/pixels
16	Effacement conditionnel du curseur
19	Fixe le seuil de passage en double vitesse

Le programme MOUSE.COM doit être chargé, soit en tapant MOUSE au clavier, soit en ayant une ligne "MOUSE" dans l'AUTOEXEC.BAT.

Lorsque l'on charge MOUSE.COM, il exécute une procédure d'initialisation et installe le programme gestionnaire de souris (driver) dans le système. Une fois installé, le driver souris reste résident jusqu'au prochain bootstrap de la machine. Si l'installation s'est déroulée avec succès, le message suivant est affiché :

"--- Installation du pilote de la souris v5.00 ---"

En cas de problème, il y a affichage de l'un des deux messages suivants :

1. **"SOURIS : Pilote de la souris déjà installé"**

   Vous aviez déjà exécuté MOUSE.COM, ou un autre utilitaire résident s'est déjà installé sur le vecteur utilisé par MOUSE.COM.

2. **"SOURIS : Souris Amstrad non trouvée"**

   Soit il y a un mauvais fonctionnement du matériel, soit votre machine n'est pas un Amstrad PC.

L'initialisation comprend les étapes suivantes :

1. La routine du ticker hardware (interruption logicielle 8) qui est appelée toutes les 54 ms est remplacée par la routine Mouse Ticker appelée toutes les 18 ms.

2. Le compteur 0 du timer 8253 est re-programmé de façon à produire une interruption toutes les 18 ms au lieu de 54. On remarque au passage que les programmes se chaînant sur cette interruption peuvent être perturbés par un appel plus fréquent dû à cette modification du rythme des interruptions.

3. Le vecteur de la routine d'interruption des boutons de la souris (Mouse button Interrupt) est implanté à l'interruption logicielle 6.

4. Les registres X et Y du mouvement de l'Amstrad PC Mouse sont initialisés à 0. On marque les boutons de la souris comme ayant été relâchés.

5. La RAM sauvegardée (NVR) est lue pour obtenir les valeurs d'échelle X et Y qui serviront à générer les déplacements du curseur à partir du mouvement de la souris en mode texte.

6. Le driver est initialisé de façon à se trouver en mode texte.

La structure générale d'un appel en assembleur d'une fonction souris du driver est celle-ci :

1. Chargement des paramètres dans les registres appropriés.
2. Appel de l'interruption logicielle 51 (033h).

Les coordonnées du curseur servant à l'appel de plusieurs fonctions sont toujours sous la forme X-Cursor (horizontal) et Y-Cursor (vertical). La valeur X-Cursor est toujours comprise entre 0 et 639, ce qui correspond à la taille d'un écran graphique haute résolution et Y-Cursor est compris entre 0 et 199. Ce système de coordonnées définit donc un écran virtuel. Lorsque l'on se trouve dans un mode présentant une plus faible résolution que 640 points, les bits de poids faible de X-Cursor sont ignorés.

Par exemple, en mode graphique 320x200, seules les valeurs paires sont signifiantes alors qu'en mode 80 colonnes, seules sont autorisées les valeurs multiples de 8 (en mode 40 colonnes, ce seront les multiples de 16). Les valeurs fournies seront arrondies aux plus proches valeurs autorisées par le mode d'affichage courant.

L'unité standard de déplacement de la souris est appelée un "mickey" et est approximativement égale à 0,127 mm (soit 1/200 ième de pouce). La fonction 15 détermine le rapport entre mickeys et pixels.

En mode texte, les mouvements de la souris provoquent l'insertion de codes clavier dans le tampon clavier. Le facteur d'échelle lu dans la NVR durant l'initialisation sert à déterminer combien de mickeys (donc l'amplitude du mouvement de la souris) seront nécessaires à provoquer l'envoi d'un code clavier dans le tampon. L'appel de toute autre fonction que les fonctions souris 0 et 2 inhibera ce mode supplémentaire (par rapport au standard PC), c'est-à-dire que ces codes clavier de mouvement du curseur ne seront plus générés. L'appel de la fonction 0 (Initialisation) ré-active ce mode texte; l'appel de la fonction 2 (Hide Cursor) ne l'affecte pas.

En mode texte, la routine d'interruption des boutons de la souris (Int 6) traduit les pressions des boutons Droit et Gauche en codes clavier (différenciés). Ces codes figurent dans la RAM NVR aux octets 29 et 30 pour le bouton gauche et 31 et 32 pour le bouton droit. La valeur NVR par défaut de ces codes est la valeur 'code ignoré' (FFh). On peut se servir du programme NVR pour affecter les valeurs souhaitées aux boutons (par exemple CR et SHIFT).

Dans les modes graphiques, la routine d'interruption des boutons de la souris traduit l'appui sur le bouton droit en appui sur la touche shift et le bouton gauche est passé à l'application de l'utilisateur comme un mouse_event (littéralement événement souris). Voir la fonction 12.

Fonction 0 : Mouse initialisation (Initialisation de la souris)

Cette fonction initialise le driver souris et retourne l'état courant du software et hardware souris.

Utilisation des registres du CPU :

Entrée : AX = 0

Sortie : AX = Etat souris
BX = Nombre de boutons

Tous les flags et les autres registres sont préservés.

Etant donné que le hardware de la souris est vérifié à la mise en route, le driver retourne toujours un état de la souris à VRAI (1). Si le driver souris n'est pas résident, AX est retourné comme FAUX (0).

Les paramètres du driver souris sont remis à leurs valeurs par défaut :

Paramètre	**Valeur**
Flag curseur	Caché (-1)
Position curseur	Centre de l'écran
Graphisme curseur	Flèche
Point d'action	-1, -1
Curseur texte	Pavé vidéo inversée
Mode émulation crayon optique	Activé
Taux Mickey / X-pixel	8
Taux Mickey / Y-pixel	16
X-Cursor Min/Max	0/639
Y-Cursor Min/Max	0/199

Les compteurs hardware X et Y de la souris sont remis à 0 comme un certain nombre de compteurs logiciels.

Le driver souris est initialisé de façon à se trouver en mode texte (et les mouvements de la souris génèrent des codes curseurs).

Fonction 1 : Show Cursor (Curseur visible)

Cette fonction incrémente le Cursor Flag et, si le flag est à 0, active l'affichage du curseur.

Utilisation des registres du CPU :

Entrée : AX = 1

Sortie : Tous les flags et les registres sont préservés.

Fonction 2 : Hide Cursor (Curseur caché)

Cette fonction décrémente le Cursor Flag.

Utilisation des registres du CPU :

Entrée : AX = 2

Sortie : Tous les flags et les registres sont préservés.

Fonction 3 : Get Mouse Position and Button Status
(Donne la position de la souris et l'état des boutons)

Cette fonction retourne l'état des boutons Gauche et Droit ainsi que la position du curseur souris.

Utilisation des registres du CPU :

Entrée : AX = 3

Sortie : BX = Etat des boutons
CX = Position X du curseur

DX = Position Y du curseur

Tous les flags et les autres registres sont préservés.

Le mot Button Status (Etat des boutons) retourné dans BX est une seule valeur entière. Les bits 0 et 1 représentent respectivement les boutons Gauche et Droit. Le bit vaut 1 si le bouton est pressé et 0 s'il ne l'est pas.

Fonction 4 : Set Mouse Cursor

Cette fonction envoie le curseur souris à la position définie par X-Cursor et Y-Cursor. Ces valeurs doivent être comprises dans les limites de l'écran virtuel. Si l'affichage ne se trouve pas en mode haute résolution, les valeurs sont arrondies aux plus proches valeurs autorisées par le mode courant.

Utilisation des registres du CPU :

Entrée : AX = 4
CX = Position X du curseur
DX = Position Y du curseur

Sortie : Tous les flags et les registres sont préservés.

Fonction 5 : Get Button Press Information
(Donne les informations sur la pression des boutons)

Cette fonction retourne l'état courant des boutons, le nombre d'appuis sur un bouton depuis le dernier appel à cette même fonction (c'est pourquoi l'on peut considérer qu'elle rend un historique) et la position X et Y au moment du dernier appui sur ce bouton.

Utilisation des registres du CPU :

Entrée : AX = 5
BX = Numéro du bouton (0 = Gauche, 1 = Droit)

Sortie : AX = Etat des boutons
BX = Nombre d'appuis sur le bouton depuis le dernier appel (0-32h)
CX = Position X lors du dernier appui
DX = Position Y lors du dernier appui

Tous les flags et les autres registres sont préservés.

Fonction 6 : Get Button Release Information
(Donne des informations sur le relâchement des boutons)

Cette fonction retourne l'état courant des boutons, le nombre de fois qu'un bouton a été relâché depuis le dernier appel à cette même fonction (c'est pourquoi l'on peut considérer qu'elle rend un historique) et la position X et Y au moment du dernier relâchement de ce bouton.

Utilisation des registres du CPU :

Entrée : AX = 6
BX = Numéro du bouton (0 = Gauche, 1 = Droit)

Sortie : AX = Etat des boutons
BX = Nombre de relâchement du bouton depuis le dernier appel (0-32h)
CX = Position X lors du dernier relâchement
DX = Position Y lors du dernier relâchement

Tous les flags et les autres registres sont préservés.

Fonction 7 : Set Minimum and Maximum X-Cursor Position
(Fixe les positions X minimum et maximum)

Cette fonction détermine les positions maximum et minimum du curseur. Par conséquent, les mouvements du curseur sont limités à la zone définie.

Utilisation des registres du CPU :

Entrée : AX = 7
CX = Position X minimum
DX = Position X maximum

Sortie : Tous les flags et les registres sont préservés.

Si le curseur se trouve hors de la zone définie lors de l'appel de la fonction, il est tout simplement ramené dans cette zone. Si le minimum est plus grand que le maximum, les deux valeurs sont échangées.

Fonction 8 : Set Minimum and Maximum Y-Cursor Position
(Fixe les positions Y minimum et maximum)

Cette fonction détermine les positions maximum et minimum du curseur. Par conséquent, les mouvements du curseur sont limités à la zone définie.

Utilisation des registres du CPU :

Entrée : AX = 7
CX = Position Y minimum
DX = Position Y maximum

Sortie : Tous les flags et les registres sont préservés.

Si le curseur se trouve hors de la zone définie lors de l'appel de la fonction, il est tout simplement ramené dans cette zone. Si le minimum est plus grand que le maximum, les deux valeurs sont échangées.

Fonction 9 : Set Graphics Cursor Block
(Définition du curseur graphique)

Cette fonction définit la forme, la couleur et le point d'action du curseur des modes graphiques.

Utilisation des registres du CPU :

Entrée : AX = 9
BX = Position relative X du point d'action (Hot Spot)
CX = Position relative Y du point d'action (Hot Spot)
DX = Pointeur vers les masques
ES = Adresse de segment des masques

Sortie : Tous les flags et les registres sont préservés.

Le point d'action a des coordonnées relatives au coin supérieur gauche du curseur. C'est à partir de lui que sont définies les coordonnées du curseur (on l'appelle aussi le Centre). Ces deux coordonnées relatives doivent être comprises entre -16 et +16.

Le dessin du curseur se compose de 2 masques. On distingue donc le masque d'écran et le masque de curseur. Ils déterminent la forme et la couleur du curseur. Le registre ES contient l'adresse de segment du tableau des masques alors que DX représente l'offset à appliquer à ES.

Les masques d'écran et de curseur sont deux tableaux de 16x16 bits disposés de manière contigue en mémoire. Le masque d'écran détermine si le pixel du curseur appartient à la forme ou au fond. Le masque de curseur détermine la manière dont le pixel sous le curseur va affecter la couleur de celui-ci. Pour créer le curseur, le logiciel de gestion de la souris effectue tout d'abord un AND entre le masque d'écran et les 256 bits qui définissent les pixels sous le curseur. Puis il effectue un XOR du masque du curseur avec le résultat du AND. La table de vérité page suivante montre la relation entre le masque d'écran, le masque de curseur et la mémoire écran qui en résulte.

Bit masque d'écran	Bit masque de curseur	Bit d'écran résultant
0	0	0
0	1	1
1	0	Inchangé
1	1	Inversé

Dans les modes graphiques 640x200 ou 640x350, chaque bit de l'écran correspond logiquement à un bit des masques et il en résulte un pixel. En modes 320x200, chaque paire de bits correspond à un pixel.

Fonction 10 : Set Text Curseur (Définition du curseur texte)

Cette fonction sélectionne le curseur hardware ou software des modes texte et lui donne ses attributs.

Utilisation des registres du CPU :

Entrée : AX = 10
BX = Type de curseur (0 = software/1 = hardware)
CX = Masque d'écran/Ligne de départ
DX = Masque de curseur/Ligne d'arrivée

Sortie : Tous les flags et les registres sont préservés.

Si le curseur texte software est choisi, CX et DX contiennent le masque d'écran et le masque de curseur. Ces masques de 16 bits passent par AND et XOR de la même manière que pour les modes graphiques et interfèrent avec les octets de caractère et d'attribut de la position (en caractères) du curseur.

Que ce soit dans les modes texte 40 ou 80 colonnes, les 16 bits d'écran de chaque caractère prennent la forme suivante :

Bits :	15	14.. 12	11	10.. 8	7..... 0
Contenu :	Cli	F	I	AvP	Caractère

où :

Cli Bit de clignotement
F Fond dont la couleur est définie par 3 bits RGB
I Bit d'intensité
AvP Avant-plan dont la couleur est définie par 3 bits RGB

Reportez-vous à la section 1.11 pour le détail sur les octets d'attribut.

Les masques d'écran et de curseur sont définis de la même manière de sorte que le curseur peut changer l'attribut d'un caractère lorsqu'il se trouve sur lui. Par exemple, un masque d'écran égal à 077FFh et un masque de curseur égal à 07700h intervertissent les couleurs de fond et d'avant-plan.

Si le curseur hardware est choisi, CX et DX définissent la ligne de départ et d'arrivée (en pixel) du curseur affiché.

Fonction 11 : Read Mouse Motion Counters
(Lecture des compteurs de déplacement de la souris)

Cette fonction retourne le nombre de mickeys horizontaux et verticaux parcourus par la souris depuis le dernier appel à cette même fonction.

Utilisation des registres du CPU :

Entrée : AX = 11

Sortie : CX = Comptage des X
DX = Comptage des Y

Tous les flags et les autres registres sont préservés.

Un mickey est l'unité standard de mouvement de la souris. Il vaut environ 1/200ième de pouce, soit 0,127 mm. C'est la fonction 15 qui détermine le ratio entre le nombre de mickeys et de pixels.

Fonction 12 : Set User Defined Subroutine Input Mask
(Définition du masque d'appel de la routine utilisateur)

Cette routine définit le masque et l'adresse d'appel d'une routine par l'interruption souris.

Utilisation des registres du CPU :

Entrée : AX = 12
Sortie : CX = Masque d'appel (Call Event Mask)
DX = Adresse d'offset de la sous-routine
ES = Adresse de segment de la sous-routine

Tous les flags et les autres registres sont préservés.

Le driver souris sonde le hardware de la souris au rythme de l'interruption ticker (environ 55 fois par seconde) et lorsqu'il détecte un des événements spécifié par le masque, il appelle votre routine. Le masque se définit comme suit :

Bit	Event
15-5	Inutilisés
5	Bouton droit relâché
3	Bouton droit pressé
2	Bouton gauche relâché
1	Bouton gauche pressé
0	Changement de position du curseur

Un bit porté à 1 indique l'événement générant l'appel.

Notez que l'appel de la fonction souris (Initialisation) inhibe tous les événements et que la fonction 12 doit alors être appelée de nouveau.

Avant d'appeler votre routine, le CPU affecte les données suivantes aux registres :

AX = Bit d'event (selon la table ci-dessus)
BX = Etat des boutons (BL = Gauche, BH = Droit)
CX = X-Cursor
DX = Y-Cursor

Fonction 13 : Light Pen Emulation Mode On
(Activation du mode émulation de crayon optique)

Dans ce mode, la souris émule le fonctionnement d'un crayon optique.

Utilisation des registres du CPU :

Entrée : AX = 13

Sortie : Tous les flags et les registres sont préservés.

Fonction 14 : Light Pen Emulation Mode Off
(Désactivation du mode émulation de crayon optique)

Utilisation des registres du CPU :

Entrée : AX = 14

Sortie : Tous les flags et les registres sont préservés.

Fonction 15 : Set Mickey/pixel Ratio (Fixe le ratio mickey/pixel)

Cette fonction définit le rapport entre l'amplitude du mouvement de la souris et celle du mouvement du curseur à l'écran.

Utilisation des registres du CPU :

Entrée : AX = 15
CX = Ratio X-mickey/pixel
DX = Ratio Y-mickey/pixel

Sortie : Tous les flags et les registres sont préservés.

Les ratios X et Y indiquent un nombre de mickey pour 8 pixels. Leur valeur doit être comprise entre 1 et 32767.

Avec une valeur de 16 mickeys pour 8 pixels, il faut déplacer la souris d'environ 16 cm pour traverser l'écran horizontalement (640 pixels). Toujours avec ce même ratio, un mouvement de souris de 5,5 cm permet de déplacer le curseur de 200 pixels.

Fonction 16 : Conditional Off (Effacement du curseur conditionnel)

Cette fonction permet de limiter la mise à jour de l'écran à une zone donnée.

Utilisation des registres du CPU :

Entrée : AX = 16
CX = Coordonnée X supérieure
DX = Coordonnée y supérieure
SI = Coordonnée X inférieure
DI = Coordonnée Y inférieure

Sortie : Tous les flags et les registres sont préservés.

Le curseur de la souris est caché durant la mise à jour de l'écran et il faut le rappeler par la fonction 11.

La fonction 16 est similaire à la fonction 2 (Hide Cursor) mais est à réserver aux applications avancées pour lesquelles une mise à jour plus rapide de l'écran est nécessaire.

Fonction 19 : Set Double Speed Threshold
(Fixe le seuil de passage en double vitesse)

Cette fonction détermine à partir de quel seuil de vitesse on va passer en déplacement rapide du curseur souris.

Utilisation des registres du CPU :

Entrée : AX = 19
DX = Seuil en Mickeys/Seconde

Sortie : Tous les flags et les registres sont préservés.

Cette fonction facilite le pointage d'images largement séparées à l'écran.

Une valeur de seuil de 0 revient à demander un seuil de 64 mickeys/seconde. Donner une valeur importante (comme 32767), inhibe le passage en double vitesse.

ANNEXE 2

CONFIGURATION DU SYSTEME SOUS MS-DOS

Le système d'exploitation MS-DOS autorise l'installation d'un grand nombre d'options de configuration durant le processus de mise en route par l'intermédiaire du fichier CONFIG.SYS qui doit figurer dans le répertoire racine du disque. Ces configurations se mettent en place par l'intermédiare des commandes suivantes :

Commande	Description
BREAK	Vérification étendue de l'occurrence d'un BREAK (CTRL-C)
BUFFERS	Nombre de tampons de secteurs
COUNTRY	Sélection des paramètres spécifiques à un pays
DEVICE	Installation de drivers de périphériques
DRIVPARM	Ecrasement des paramètres d'origine d'un périphérique
FCBS	Nbre de fichier ouverts par File Control Block
FILES	Nbre maximum d'identificateurs de fichiers ouverts simultanément
LASTDRIVE	Lettre du dernier lecteur accessible
SHELL	Désignation du processeur de commande de niveau supérieur
STACKS	Ecrasement des paramètres par défaut de la pile de ressources

CONFIG.SYS est un fichier au format ASCII qui peut être créé avec n'importe quel éditeur. RPED convient très bien.

2.1 LA COMMANDE BREAK

Cette commande met en oeuvre ou inhibe la vérification étendue de l'occurrence d'un BREAK par MS-DOS. Normalement, MS-DOS vérifie qu'un [Ctrl][C] ait été envoyé lorsqu'il lit les données du clavier et lorsqu'il écrit à l'écran ou à l'imprimante. Mettre en oeuvre la vérification étendue revient à demander à MS-DOS de tenir compte aussi des BREAK intervenant durant d'autres fonctions et en particulier les accès disques. La syntaxe de la commande BREAK est la suivante :

BREAK=[ON]

ou

BREAK=[OFF]

En l'absence de cette commande ou d'un paramètre, le mode OFF est choisi implicitement.

2.2 LA COMMANDE BUFFERS

Cette commande permet de fixer le nombre de tampons des opérations sur disque alloués au démarrage de MS-DOS. Un tampon disque (disk buffer) est une zone mémoire où MS-DOS conserve lés données lues ou à écrire sur disque tant que la quantité totale de données n'est pas un multiple exact de la taille d'un secteur.

La syntaxe de la commande BUFFERS est :

BUFFERS=n

où n est un nombre entre 2 et 255. Si la commande BUFFERS est absente, MS-DOS affecte 2 tampons. Le nombre de tampons reste constant depuis le lancement de la machine jusqu'à ce qu'elle soit arrêtée ou relancée. En général, on recommande une valeur comprise entre 10 et 20 tampons pour des applications standard (traitements de textes, tableurs) afin de diminuer les temps d'accès disques. Si vous avez créé beaucoup de sous-répertoires, une valeur de 30 environ peut mieux convenir.

Mais étant donné que chaque tampon occupe 0,5 K de mémoire vive, il y a un compromis à rechercher entre les performances des opérations sur disques et l'encombrement mémoire. Les tampons alloués au dessus de 40 ne servent à rien.

Les manuels de certaines applications recommandent un nombre de tampons minimum. Consultez-les en cas de doute.

2.3 LA COMMANDE COUNTRY

La commande COUNTRY permet de choisir le mode d'affichage de certaines informations (date, heure, etc...).

La syntaxe de cette commande est :

COUNTRY=nnn

où nnn est le code du pays. Vous le trouverez en annexe 3. Ce code correspond au préfixe téléphonique du pays lorsque l'on fait un appel international. Notez que seuls sont affectés les paramètres décrits dans le tableau de l'Annexe 3 mais que d'autres éléments peuvent intervenir tels que la configuration des liaisons d'option, les choix faits avec N-Utility, KEYBxx, et les variantes nationales du système d'exploitation.

2.4 LA COMMANDE DEVICE

Cette commande permet d'installer des drivers de périphériques figurant dans un répertoire spécifié.

La syntaxe de la commande DEVICE est :

DEVICE=<unité><chemin>

Le fichier spécifié est chargé et exécuté. Le driver peut alors procéder à son initialisation et à celle du système. Reportez-vous au manuel de référence technique de MS-DOS pour trouver la manière d'écrire des drivers de périphériques.

Votre disque MS-DOS (Disk 1) contient deux drivers de périphériques : DRIVER.SYS et RAMDRIVE.SYS qui permettent respectivement de configurer des unités de mémoire de masse externes et d'installer un disque virtuel.

Si vous avez l'intention de vous servir des codes d'échappement décrits dans le manuel du PC1640, vous devez ajouter à votre CONFIG.SYS :

```
DEVICE=ANSI.SYS
```

de cette manière, vous pourrez profiter des nombreuses options de contrôle de l'écran et du clavier offertes par l'émission de codes ANSI. Reportez-vous à votre Manuel Utilisateur du PC 1640 (livre 1), annexe 4 pour obtenir la liste de ces codes.

2.4.1 DRIVER.SYS

DRIVER.SYS est un gestionnaire (driver) de périphériques installable destiné à définir les caractéristiques de lecteurs externes. Pour installer DRIVER.SYS, ajoutez la ligne suivante à votre ficher CONFIG.SYS :

```
DEVICE=DRIVER.SYS /D:dd {/C}{/F:ff}{/H:hh}{/N}{/S:ss}{/T:tt}
```

où l'on trouve :

/D:dd dans lequel est le numéro de lecteur (de 0 à 127 pour un lecteur de disquette ou de 128 à 255 pour un disque dur).

et où on peut rajouter optionnellement :

/C indique que le support de blocage de porte est demandé.
/F:ff dans lequel ff spécifie le type d'unité :

0 = disquette 5,25 pouces, 320/360 K
1 = disquette 5,25 pouces, 1,2 Méga octets
2 = disquette 3,5 pouces, 720 K
3 = disquette 8 pouces simple densité

4 = disquette 8 pouces double densité
5 = disque dur
6 = Streamer ou lecteur de cassettes
7 = Autre

/H:hh dans lequel hh est le numéro de tête maximum (1-99)
/N indique une unité inamovible
/S:ss dans lequel ss est le nombre de secteurs par piste (1-99)
/T:tt dans lequel tt est le nombre de pistes par face (1-999)

2.4.2 RAMDRIVE.SYS

RAMDRIVE.SYS est un device driver installable qui permet de se servir d'un partie de la mémoire comme s'il s'agissait d'un lecteur de disque. On appelle cela un RAM disque ou disque virtuel.

Si vous possédez une extension de mémoire répondant aux spécifications LIM (Lotus (R), Intel (R), Microsoft (R)), vous pouvez y implanter un ou plusieurs disques virtuels. Sinon, RAMDRIVE.SYS implantera le RAM disque dans la mémoire utilisateur.

Pour installer le disque virtuel, faites figurer l'instruction suivante dans votre CONFIG.SYS :

DEVICE=RAMDRIVE.SYS {kkkk}{ssss}{dddd}{/A}

où :

kkkk est la taille du disque virtuel en Kilo octets. La valeur par défaut est 64 et la taille minimum est 16 K (ou NVR pour utiliser la valeur de la NVR).

ssss est la taille de secteur. On peut indiquer 64 (minimum), 128, 256 ou 512. La valeur par défaut est 128.

dddd est le nombre d'entrées dans le catalogue du répertoire racine. Cette valeur doit être comprise entre 2 et 1024 (défaut = 64).

/A indique qu'une carte d'extension mémoire répondant aux spécifications LIM est présente et doit être utilisée pour l'implantation du disque virtuel. Cette option est incompatible avec /E.

Il existe un autre paramètre optionnel à cette commande. Il s'applique aux machines dont l'architecture s'organise autour d'un 80286 et dont la mémoire vive dépasse les 1 méga octets :

/E indique que la mémoire étendue (au dessus de 1 Mo) doit être utilisée. Cette option est incompatible avec /A.

2.5 LA COMMANDE DRIVPARM

Cette commande DRIVPARM permet d'allouer de nouvelles valeurs aux paramètres du driver de périphérique d'une unité de disques logique en écrasant les valeurs d'origine.

Syntaxe :

DRIVPARM=/D:dd {/F:ff}{/T:tt}{/S:ss}{/N}{/C}{/H:hh}

où l'on trouve :

/D:dd dans lequel est le numéro de lecteur.

et où on peut rajouter optionnellement :

/T:tt dans lequel tt est le nombre de pistes par face (1-999)
/S:ss dans lequel ss est le nombre de secteurs par piste (1-99)
/H:hh dans lequel hh est le numéro de tête maximum (1-99)
/C indique que le support de blocage de porte est demandé.
/N indique une unité inamovible
/F:ff dans lequel ff spécifie le type d'unité :

0 = disquette 5,25 pouces, 320/360 K
1 = disquette 5,25 pouces, 1,2 Méga octets
2 = disquette 3,5 pouces, 720 K
3 = disquette 8 pouces simple densité

4 = disquette 8 pouces double densité
5 = disque dur
6 = Streamer ou lecteur de cassettes
7 = Autre

Cette commande permet donc de modifier les paramètres relatifs à un périphérique déjà installé. Cette information serait utilisée par les commandes DOS pour, par exemple, le formatage d'une disquette ou sa copie (FORMAT et COPY) lors de l'écriture du catalogue et de la FAT (File allocation table). Ces informations de la FAT détermineront ensuite les caractéristiques du périphérique.

2.6 LA COMMANDE FCBS

La commande FCBS vous permet de spécifier le nombre de blocs de contrôle de fichiers (File Control Blocks) réservés par le système et par conséquent le nombre de fichiers pouvant être ouverts simultanément.

La syntaxe de cette commande est :

FCBS=x,y

où x est le nombre de FCB (de 1 à 255) à allouer et y le nombre de FCB protégés d'une fermeture automatique lorsque le programme ouvre plus de x fichiers. Les y premiers fichiers ouverts seront protégés. Lorsqu'il doit fermer automatiquement un fichier, MS-DOS choisit celui qu'il a ouvert le moins récemment parmi ceux qui ne sont pas protégés.

Il faut évidemment choisir un y plus petit que x.

Si la commande FCBS ne figure pas dans CONFIG.SYS, les valeurs par défaut sont adoptées : x=4 et y=0.

2.7 LA COMMANDE FILES

Cette commande détermine le nombre maximum d'identificateurs de fichiers pouvant être affectés simultanément. Lorsqu'un programme ouvre le canal d'un fichier ou d'un périphérique, celui-ci reçoit un identificateur (handle) qui sert au programme à le désigner.

La syntaxe de la commande FILES :

FILES=n

où n est le nombre d'identificateurs compris entre 8 et 255. Lorsque l'on ne fait pas figurer cette commande dans CONFIG.SYS, la valeur par défaut est prise en compte : 8 fichiers. Une valeur supérieure à 20 ne sert à rien.

2.8 LA COMMANDE LASTDRIVE

LASTDRIVE indique la lettre maximale à affecter à une unité de disque.

La syntaxe de cette commande est la suivante :

LASTDRIVE=d

où d est une lettre entre "A" et "Z" (elle peut être indifféremment majuscule ou minuscule). Quand cette lettre à un rang inférieur à celui de la dernière unité de disque effectivement (physiquement) présente, la commande est ignorée. La valeur par défaut est "E".

2.9 LA COMMANDE SHELL

La commande SHELL sert à lancer (au niveau supérieur des appels) un interpréteur de commandes autre que COMMAND.COM.

La commande a la forme :

SHELL={lecteur:}<chemin> {param1{param2...{paramn}}}

Cette commande est utilisée en conjonction avec quelques grands logiciels qui fournissent leur propre interpréteur de commandes. Le manuel technique MS-DOS contient les informations nécessaires au développement d'un interpréteur de commandes.

2.10 LA COMMANDE STACKS

Cette commande vous permet d'écraser par de nouvelles valeurs les paramètres de gestion de piles du DOS. Chaque fois que se produit une interruption hardware, MS-DOS lui alloue une pile parmi la liste des piles. Lorsque le processus d'interruption est achevé, MS-DOS "rend" la pile à la liste des piles disponibles.

La syntaxe de STACKS est :

STACKS=<nombre de piles>,<taille d'une pile en octets>

S'il n'y a pas de ligne STACKS dans CONFIG.SYS, MS-DOS adopte les valeurs par défaut, équivalentes à STACKS=9,128. Cependant, cela peut se révéler insuffisant si vous avez installé beaucoup de périphériques générant des interruptions (réseau local, co-processeur 8087, disque dur). Si vous obtenez fréquemment des erreurs du type *"Internal Stack Failure"* ou (parfois) *"Divide Overflow"*, il peut être intéressant de tenter d'augmenter ces valeurs. De même que pour d'autres commandes (BUFFERS, FCBS), STACKS prend la mémoire allouée dans la zone utilisateur. Il y a donc un compromis à rechercher.

Le nombre de piles peut être compris entre 8 et 64 et leur taille peut varier de 32 à 512 octets.

ANNEXE 3

MODES D'AFFICHAGES MS-DOS DEPENDANT DU PAYS.

Pays	Num	fd	sd	sh	fh	sym	fn	cs	sm	sed	sl
Australie	061	1	-	:	1	$	0	2	,	.	,
Belgique	032	1	/	:	1	F	3	2		,	;
Canada	002	2	-	:	1	$	3	2		,	;
Danemark	045	1	/	.	1	DKR	3	2	.	,	;
Finlande	358	1	-	:	1	MK	3	2		,	;
France	033	1	/	:	1	F	3	2		,	;
Allemagne	049	1	.	.	1	DM	0	2	.	,	;
Italie	039	1	/	:	1	Lit.	1	0	.	,	;
Israël	972	1	/	:	1	ö	2	2	,	.	,
Moyen Orient	785	1	/	:	1	$	3	3	.	,	;
Pays Bas	031	1	-	:	1	ƒ	0	2	.	,	;
Norvège	047	1	/	.	1	kr	2	2	.	,	;
Portugal	351	1	/	:	1	$	4	2	.	,	;
Espagne	034	1	/	:	1	Pt	3	2	.	,	;
Suède	046	2	-	.	1	SEK	2	2	.	,	;
Suisse	041	1	.	.	1	Fr	2	2	,	.	,
Gr. Bretagne	044	1	-	:	1	£	0	2	,	.	,
Etats Unis	001	0	-	:	0	$	0	2	,	.	,

où :

Num = Numéro de code du pays (correspond à l'indicatif téléphonique international).

fd = Format de la date : 0 = format américain MM/JJ/AA, 1 = format européen JJ/MM/AA, 2 = format japonais AA/MM/JJ.

sd = Séparateur de date.

sh = Séparateur de l'heure.

fh = Format de l'heure : 0 = sur 12 heures, 1 = sur 24 heures.

sym = Symbole monétaire.

fn = Format numéraire : si le bit 0 est égal à 1, le symbole monétaire précède le nombre ; s'il est égal à 0, le symbole suit le nombre. Bits 1 et 2 = nombre d'espaces entre le nombre et le symbole.

cs = Nombre de chiffres décimaux significatifs en format numéraire.

sm = Séparateur des milliers.

sed = Séparateur décimal.

sl = Séparateur d'une liste de données.

ANNEXE 4

LES CONNEXIONS RS232C

Pour bien comprendre les problèmes liés aux communications RS232, il faut se souvenir que tout dispositif informatique disposant d'une interface série peut être classifié soit comme un modem, soit comme un terminal. Pratiquement, un modem est un moyen d'allonger la distance séparant deux terminaux connectés (via une ligne téléphonique). La figure 1 est l'illustration schématisée d'une connexion terminal à terminal par l'intermédiaire de modems.

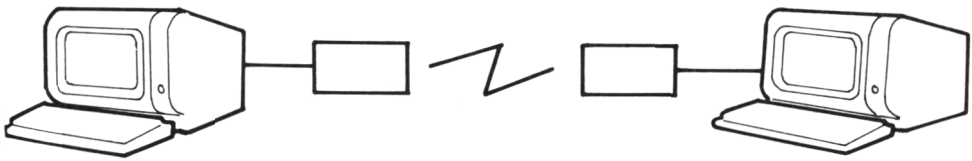


FIGURE 1 :

Connexion de terminaux idéalisée

Le connecteur le plus fréquemment utilisé en matière de liaisons série possède 25 broches sur lesquelles 7 seulement sont nécessaires dans la plupart des cas de figure. En principe, la connexion de 2 modems se fait par l'intermédiaire d'un câblage droit (dit one-to-one en anglais), c'est-à-dire que la broche 1 d'un modem est connectée à la broche 1 de l'autre, la 2 à la 2, etc... jusqu'à la broche 25.

Avec un câble de ce type, le transfert d'informations se passe comme ceci :

En suivant le parcours du signal de gauche à droite (figure 2), les caractères provenant du clavier (keyboard) sont envoyés sous forme série par la broche 2 du terminal de gauche vers la broche 2 du modem de gauche (broche marquée Transmit Data, soit TD en nomenclature RS232). Le modem de gauche envoie ces caractères vers le modem de droite par une ligne téléphonique. Les caractères sont reçus par la broche 3 du modem de droite (pin Receive Data RD) et renvoyés vers la broche 3 du terminal de droite. A réception de ces caractères, le terminal de droite les affiche à l'écran.

Notez bien que les termes *"transmit data"* (transmission de données) et *"receive data"* (réception de données) n'ont leur signification que par rapport au terminal et non au modem.

Le cheminement des données de la gauche vers la droite tel que nous venons de le décrire s'applique aux cablâges reliant les broches du même nom (portant le même numéro). Chaque terminal envoie bien les données par TD et en reçoit bien par RD. Cette disposition est parfaitement symétrique et n'engendre pas de confusion quant à l'utilisation des broches et la direction du flux de données.

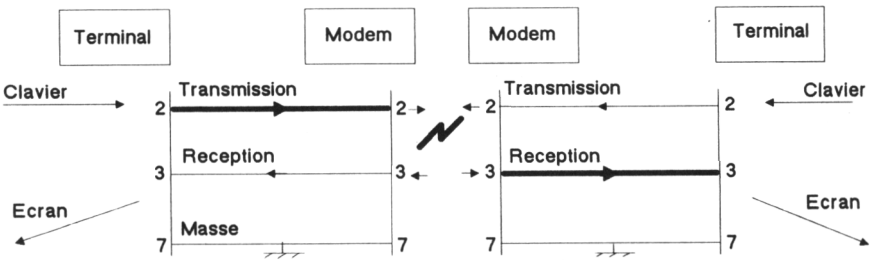


FIGURE 2

Les choses se compliquent lorsque l'on veut connecter directement deux terminaux sans passer par des modems. On ne peut pas connecter la broche 2 de l'un à la broche 2 de l'autre puisque les deux terminaux vont transmettre vers la broche de l'autre terminal qui justement est destinée elle aussi à transmettre. Il en est de même pour la réception. La solution évidente consiste à croiser les fils des broches 2 et 3 de façon à ce que la broche émettrice (TD) du terminal gauche soit bien reliée à la broche réceptrice du terminal droit. On appelle ce type de câble *"Null-modem"* puisqu'il permet de se dispenser de modem. On dit aussi souvent que le câble est *"croisé"*. Les broches 7 (la masse) restent bien évidemment reliées entre elles.

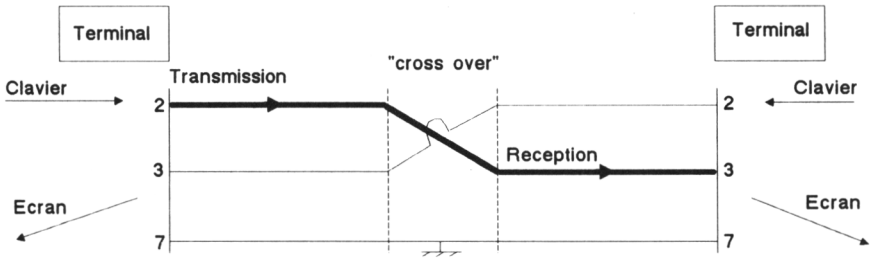


FIGURE 3

Naturellement, l'Amstrad PC 1640 avec son interface RS232C est considéré comme un terminal et sera donc relié à un modem par un câble droit.

Un câble Null-modem sera nécessaire pour relier votre Amstrad à un autre terminal. Nous entendons par terminal un autre ordinateur Amstrad avec une RS232C (ou tout autre PC ou machine haut de gamme), un moniteur conventionnel (VDU), une imprimante à interface série ou un périphérique série quelconque.

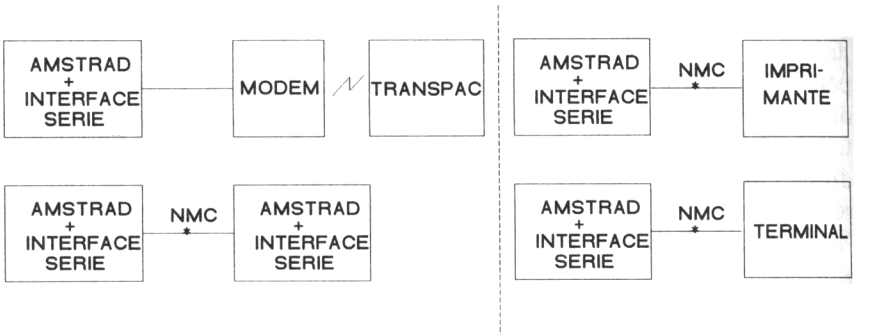


FIGURE 4

Mais il faut remarquer que si en matière de liaison entre deux unités centrales de type PC on ne risque guère de surprises, certains fabriquants de périphériques ou d'ordinateurs portables ont déjà croisé le cablage du connecteur RS232 pour "faciliter" les connexions. Il faudra alors revenir au câble droit.

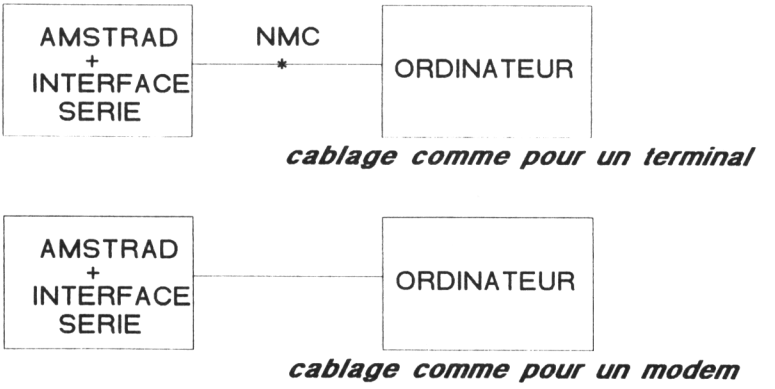


FIGURE 5

Si le monde était parfait, il serait possible de reconnaître le type des câblages puisqu'en principe les câbles droits sont munis de connecteurs mâles et les câbles croisés de connecteurs femelles. Malheureusement cette convention n'est pas respectée et bon nombre de fabriquants de "terminaux" et d'imprimantes mettent de toutes façons des connecteurs femelles sur leur matériel essentiellement pour des considérations de sécurité électrique.

En cas de doute, on peut consulter les manuels des matériels afin de découvrir le câblage prévu. Si la broche 2 est marquée TRANSMIT, l'équipement peut être considéré comme un terminal, si elle est marquée RECEIVE l'équipement est câblé comme un modem.

Contrôle hardware du flux des données

La connexion simplifiée décrite ci-dessus ne permet aucun contrôle du flux des données. En pratique, il est souhaitable que le dispositif receveur possède le contrôle de la transmission pour éviter que l'émetteur ne le submerge de données qu'il n'aura pas eu le temps de traiter parce que son travail est plus lent que celui de l'émetteur. De plus, si l'émetteur a pour une raison quelconque des doutes sur les données qu'il envoie, il doit avoir la possibilité d'interrompre le travail du récepteur. Dans le cas d'une connexion terminal à un modem, quand le terminal est prêt à envoyer, il active sa broche 4 dite RTS (Request To Send). Lorsque le modem est prêt à recevoir, il active la broche 5 (CST Clear To Send). Ce n'est qu'à partir du moment où CTS est activée que le terminal va se mettre à envoyer et il ne le fera que tant que CST sera activée. Le modem peut donc contrôler le débit des données en se servant de CTS.

Quand le modem considère que les données qu'il veut envoyer sont correctes, il active la broche 8 (DCD Data Carrier Detect). Quand le terminal est prêt à recevoir, il active la broche 20 : Data Terminal Ready (DTR). Le modem ne transmettra que si DTR est active et donc le terminal peut contrôler le flux grâce à DTR.

Il existe encore deux signaux à présenter. L'un (broche 22) est RI (Ring Indicator) qui permet au modem d'annoncer au terminal que le téléphone est en train de sonner (à ce moment, le logiciel du terminal est sensé se réveiller). L'autre signal est sur la broche 6, il s'agit de DSR (Data Set Ready). Ce signal est ignoré par le côté receveur de la RS232C ; dans la plupart des cas, le modem lancera ce signal en même temps que DCD et l'on peut donc ignorer DSR sans craindre de perte d'informations.

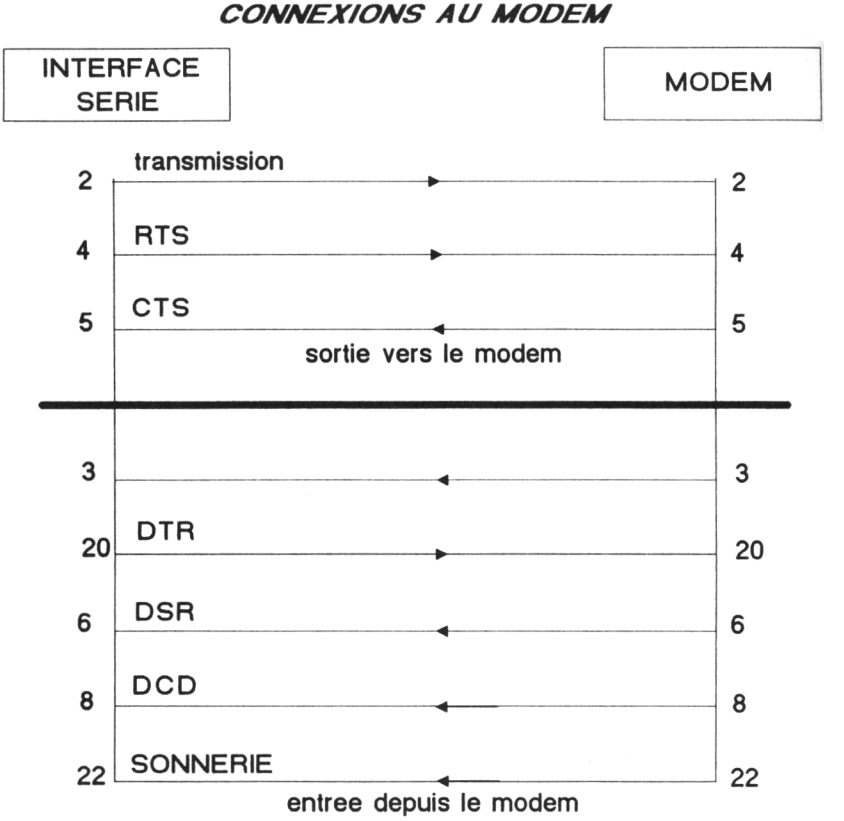


FIGURE 6

Dans le cas de connexion terminal à terminal, il faut se servir du câble Null-modem qui comprend de plus les connexions des broches 2, 3 et 7 telles que nous les avons décrites précédemment. Ce câble, lorsqu'il est complet, intervertit les broches 4 et 8 (RTS et DCD qui permettent de signaler qu'on est prêt à émettre) et les broches 20 et 5 (DTR et CTS les broches signalant qu'on n'est pas prêt). Pour plus de sûreté, et quand ça ne marche toujours pas, on peut relier les broches 6 (DSR) et 8 (DCD) du même côté si l'on tombe sur un terminal "chipoteur" qui veut un DSR en plus du DCD.

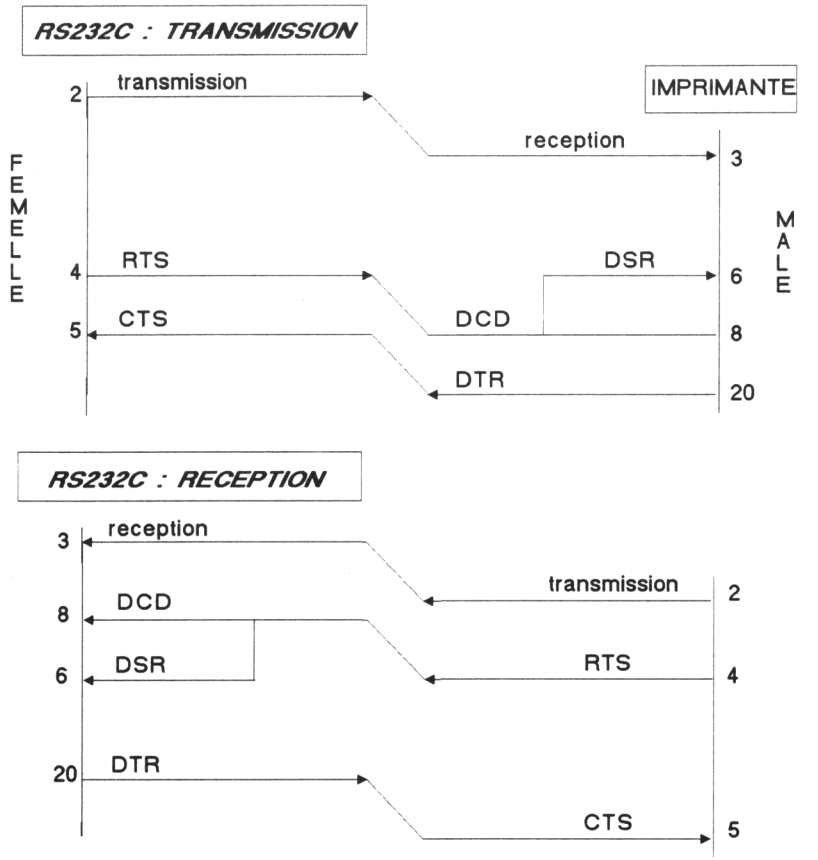


FIGURE 7

Dans les cas désespérés, il y a une école qui dit que dans le cas de liaisons Null-modem, et contrairement aux modem qu'elle remplace, les deux parties sont TOUJOURS prêtes à émettre. On s'arrange donc pour générer DCD (et DSR) en permanence en leur reliant le RTS du même côté du câble.

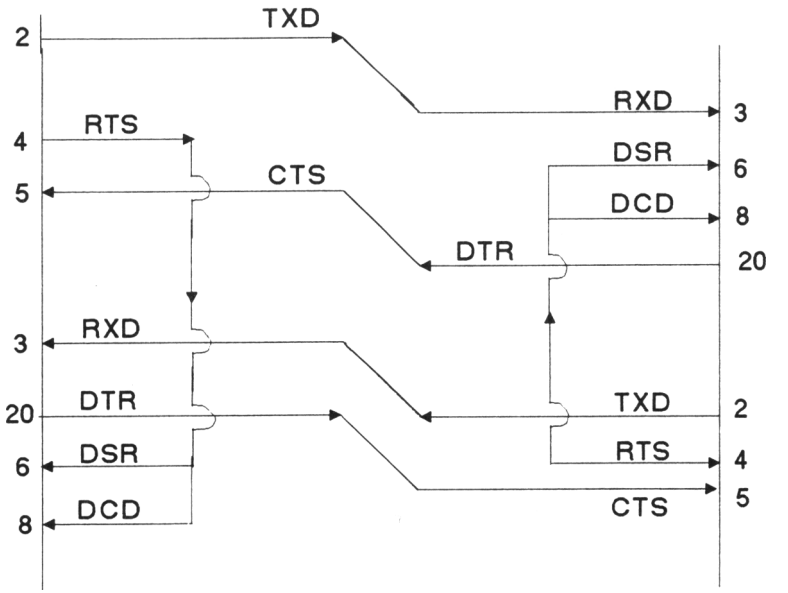


FIGURE 8

Câble Null-modem recommandé

Finalement, si l'on sait que le débit de la transmission de l'un des deux terminaux n'est pas stoPpable (par exemple, une personne tapant au clavier), ou qu'il est si lent ou si peu fréquent (par exemple, les caractères de handshake, c'est-à-dire d'accord, XON et XOFF envoyés par une imprimante) qu'il n'y a pas de danger de submerger le récepteur, on peut autoriser la transmission de façon permanente en reliant la broche 5 (CST) à la broche 4 (RTS). On envoie alors un signal de "prêt à recevoir" en permanence. Dans ces conditions, il peut être plus simple pour le terminal émetteur d'ignorer l'état de CTS.

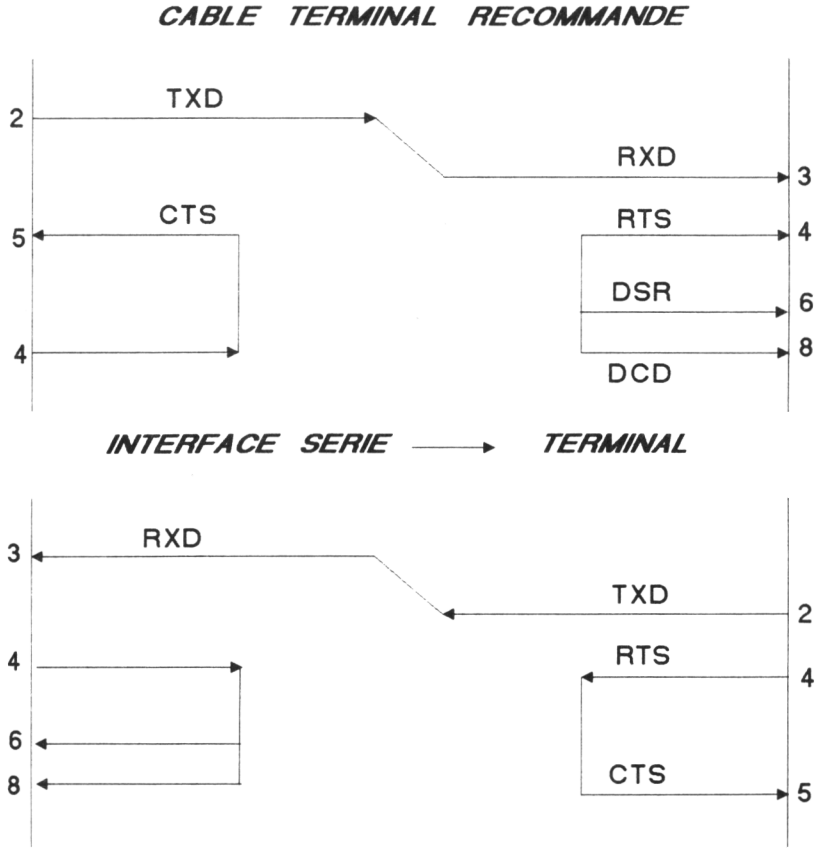


FIGURE 9

ANNEXE 5

CABLAGE PRINTER LEAD (PL-2)

Connecteurs :

1. Ordinateur : connecteur D-plug parallèle Centronics 25 points.

2. Imprimante : connecteur 36 points IEEE-488 plug.

Câblage :

Nom de la ligne	Connecteur ordinateur	Connecteur imprimante
-Strobe	1	1
Data Bit 0	2	2
Data Bit 1	3	3
Data Bit 2	4	4
Data Bit 3	5	5
Data Bit 4	6	6
Data Bit 5	7	7
Data Bit 6	8	8
Data Bit 7	9	9
-Ack	10	10
Busy	11	11
PO	12	12
Slct Out	13	13
-Auto Fd	14	14
-Error	15	32
-Reset	16	31
-Slct In	17	36
Masse	18	19
Masse	19	20
Masse	20	21

Nom de la ligne	Connecteur ordinateur	Connecteur imprimante
Masse	21	22
Masse	22	23
Masse	23	24
Masse	24	25
Masse	25	26
Masse		27
Masse		28
Masse		29
Masse		30
Masse		33
NC		15
NC		16
NC		17
NC		18
NC		34
NC		35

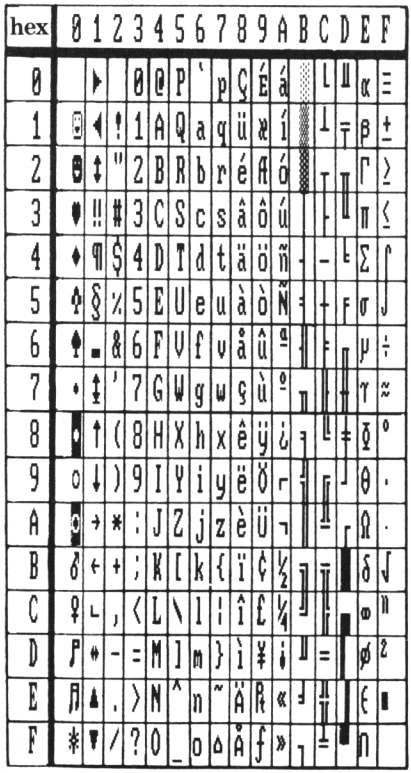
ANNEXE 6

DISTRIBUTION DE L'ALIMENTATION DU PC 1640

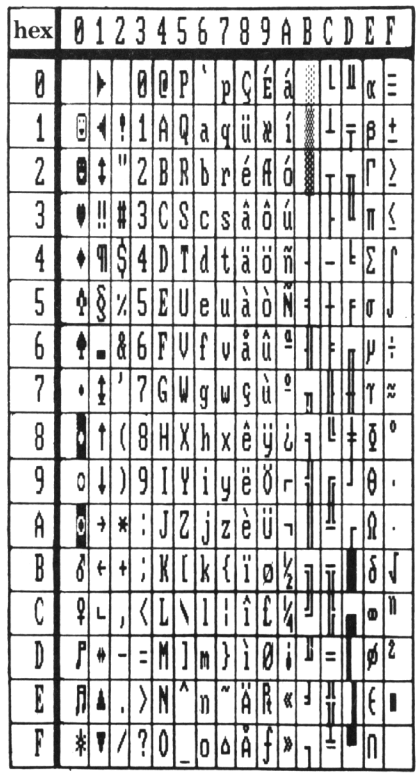
Composante principale	+5V	+12V	-5V	-12V
Carte mère et clavier	2,10	0,15	0,00	0,00
Lecteur de disquettes	0,40	1,00	0,00	0,00
Disque dur	1,10	0,80	0,00	0,00
Connecteurs d'extensions	2,10	0,15	0,10	0,15
Total	5,70	2,10	0,10	0,25

ANNEXE 7

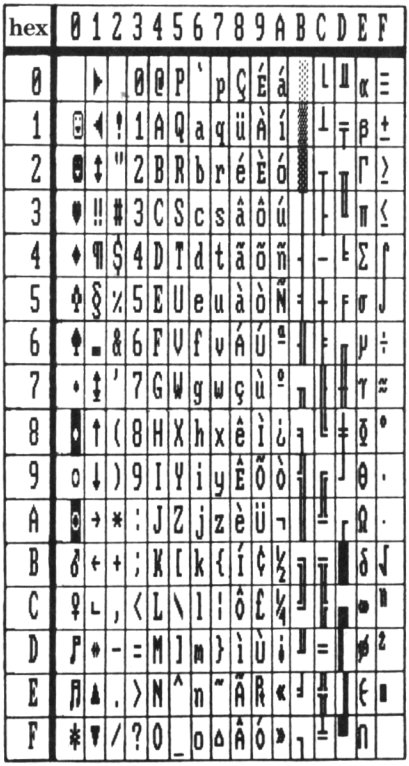
JEUX DE CARACTERES EN ROM



JEUX DE CARACTERES ANGLAIS

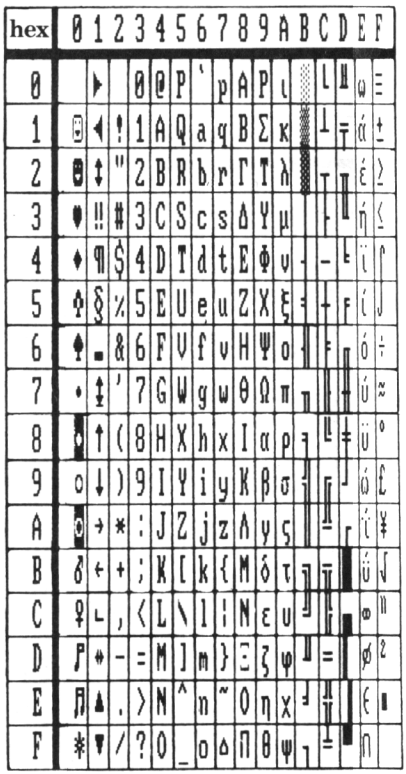


JEUX DE CARACTERES DANOIS



hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		▶		0	@	P	`	p	Ç	É	á	▒	L	╨	α	≡
1	☺	◀	!	1	A	Q	a	q	ü	À	í	▓	┴	═	ß	±
2	☻	↕	"	2	B	R	b	r	é	È	ó	▓	┬	╥	Γ	≥
3	♥	‼	#	3	C	S	c	s	â	ô	ú	▓	├	╫	π	≤
4	♦	¶	$	4	D	T	d	t	ã	õ	ñ	▓	─	╪	Σ	⌠
5	♣	§	½	5	E	U	e	u	à	ò	Ñ	▓	┼	┘	σ	⌡
6	♠	▬	&	6	F	V	f	v	Á	Ú	ª	▓	╞	┌	µ	÷
7	•	↨	'	7	G	W	g	w	ç	ù	º	▓	╟	╫	τ	≈
8	◘	↑	(	8	H	X	h	x	ê	Ì	¿	▓	╚	╪	Φ	°
9	○	↓	)	9	I	Y	i	y	Ê	Õ	ò	▓	╔	┘	Θ	·
A	◙	→	*	:	J	Z	j	z	è	Ü	¬	▓	═	┌	Ω	·
B	♂	←	+	;	K	[	k	{	Í	¢	½	▓	╩	╥	δ	√
C	♀	∟	,	<	L	\	l	¦	ô	£	¼	▓	╦	▄	∞	ⁿ
D	♪	↔	-	=	M	]	m	}	ì	Ù	¡	▓	═	▌	φ	²
E	♫	▲	.	>	N	^	n	~	Ã	₧	«	▓	╬	▐	ε	■
F	☼	▼	/	?	O	_	o	⌂	Â	Ó	»	▓	╧	▀	∩	

JEUX DE CARACTERES PORTUGAIS

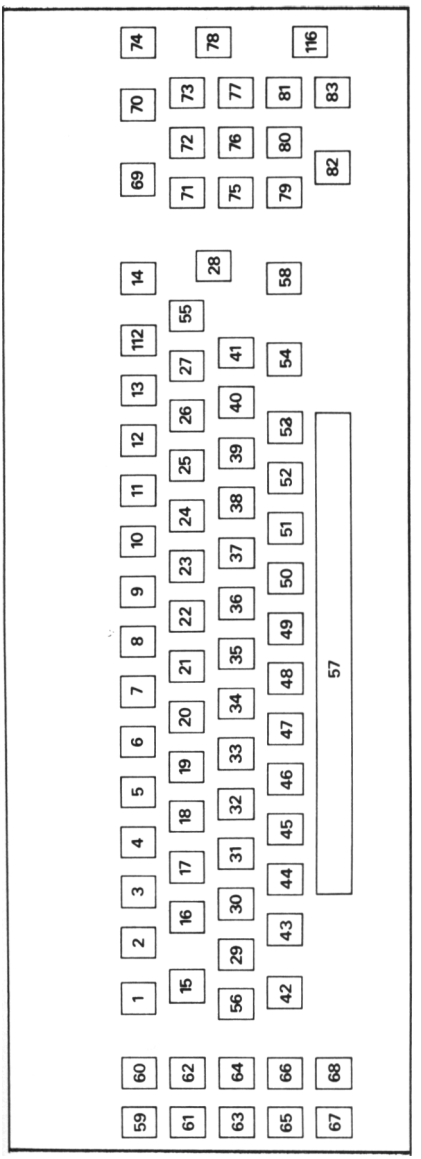


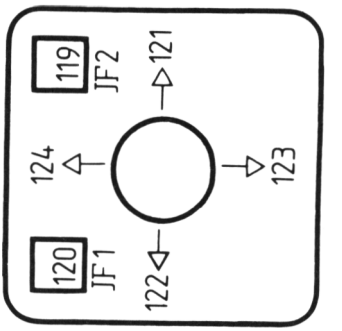
hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		▶		0	@	P	`	p	Α	Ρ	ι	▒	L	╨	ω	≡
1	☺	◀	!	1	A	Q	a	q	Β	Σ	κ	▓	┴	═	ά	±
2	☻	↕	"	2	B	R	b	r	Γ	Τ	λ	▓	┬	╥	έ	≥
3	♥	‼	#	3	C	S	c	s	Δ	Υ	μ	▓	├	╫	ή	≤
4	♦	¶	$	4	D	T	d	t	Ε	Φ	υ	▓	─	╪	ϊ	⌠
5	♣	§	½	5	E	U	e	u	Ζ	Χ	ξ	▓	┼	┘	ί	⌡
6	♠	▬	&	6	F	V	f	v	Η	Ψ	ο	▓	╞	┌	ό	÷
7	•	↨	'	7	G	W	g	w	Θ	Ω	π	▓	╟	╫	ύ	≈
8	◘	↑	(	8	H	X	h	x	Ι	α	ρ	▓	╚	╪	ϋ	°
9	○	↓	)	9	I	Y	i	y	Κ	β	σ	▓	╔	┘	ώ	∙
A	◙	→	*	:	J	Z	j	z	Λ	γ	ς	▓	═	┌	Ώ	·
B	♂	←	+	;	K	[	k	{	Μ	δ	τ	▓	╩	╥	ΰ	√
C	♀	∟	,	<	L	\	l	¦	Ν	ε	υ	▓	╦	▄	∞	ⁿ
D	♪	↔	-	=	M	]	m	}	Ξ	ζ	φ	▓	═	▌	ϕ	²
E	♫	▲	.	>	N	^	n	~	Ο	η	χ	▓	╬	▐	ε	■
F	☼	▼	/	?	O	_	o	⌂	Π	θ	ψ	▓	╧	▀	∩	

JEUX DE CARACTERES GRECS

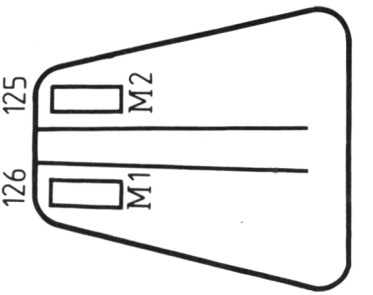
ANNEXE 8

CODES DES TOUCHES DU CLAVIER





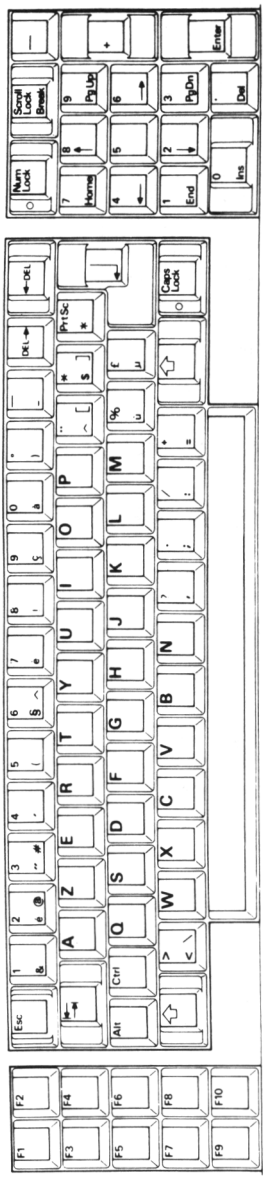
JOYSTICK



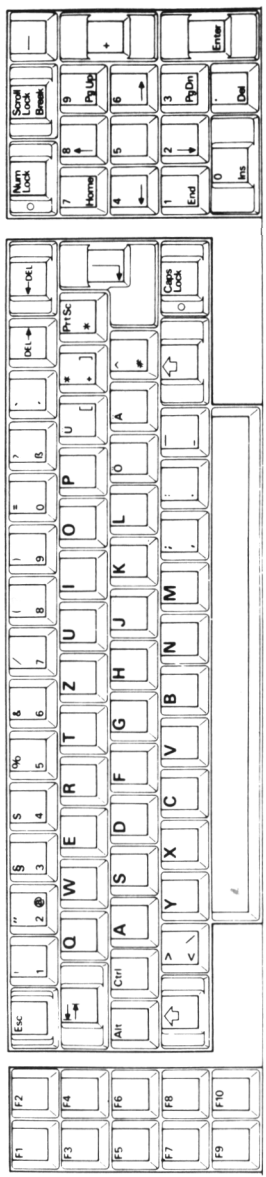
SOURIS

ANNEXE 9

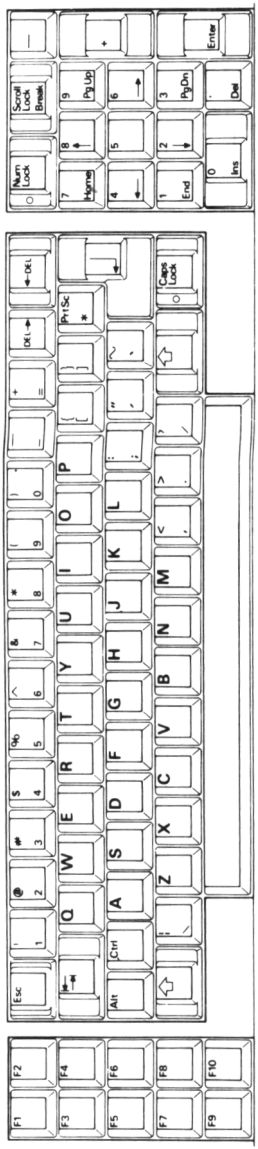
DISPOSITION DES TOUCHES



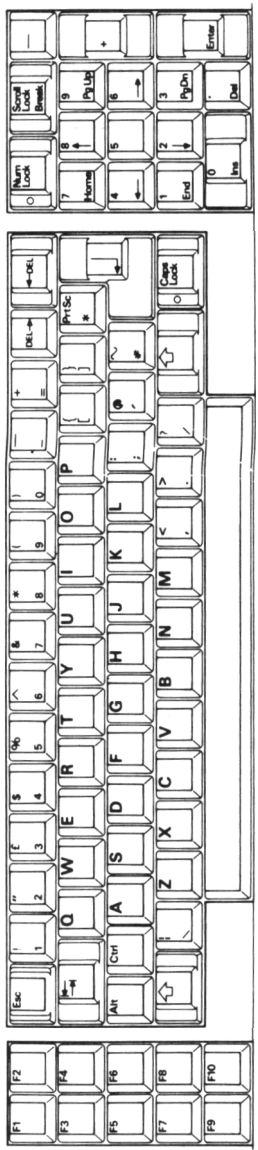
Clavier français



Clavier allemand

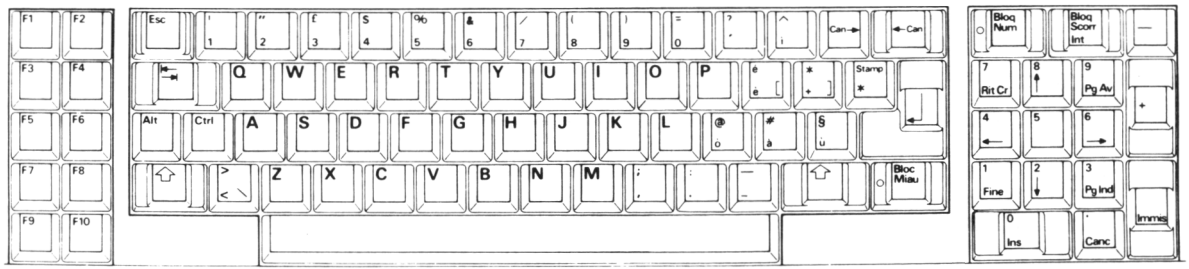


Clavier US



Clavier GB

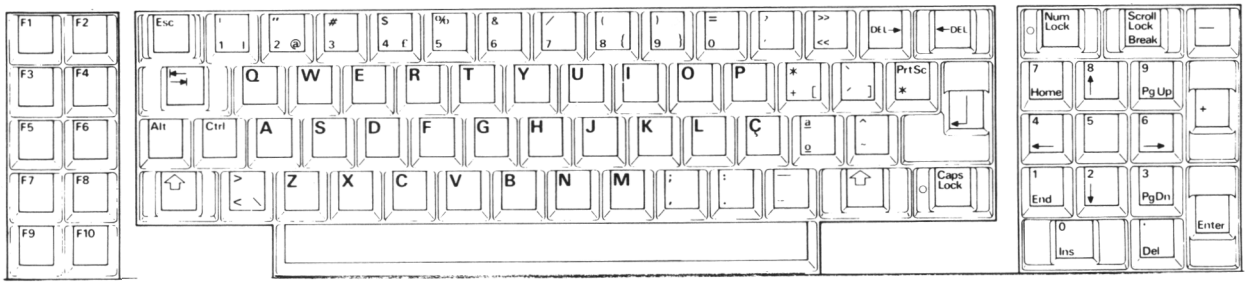
Clavier italien



Clavier espagnol

Clavier suédois

Clavier danois



Clavier portugais

Clavier grec

Clavier norvégien

ANNEXE 10

L'EDITEUR DE LIENS (MS-LINK)

10.1 INTRODUCTION

Cette annexe se propose d'expliquer l'utilisation de MS-LINK. Nous nous recommandons d'en finir la lecture avant de commencer à vous servir du programme.

L'éditeur de liens (linker) fourni avec MS-DOS (appelé LINK.EXE) est un programme qui :

1. Lie plusieurs modules objets créés séparément en un module relogeable unique, c'est-à-dire un programme .EXE que vous pourrez lancer (exécutable).

2. Cherche les définitions ou les références externes non résolues dans des fichiers bibliothèques.

3. Résoud les références croisées externes.

4. Produit un listing où figurent à la fois les résolutions de références externes et les messages d'erreurs.

10.2 SURVOL DE MS-LINK

Lorsque vous écrivez un programme, vous rédigez un code source. Ce code source est traité par un compilateur qui produit des modules objets. Ces modules objets doivent passer à leur tour par l'éditeur de liens pour produire des fichiers en langage machine que l'ordinateur puisse comprendre directement. Ce sont ces fichiers-là qu'il peut exécuter.

Vous pouvez vouloir lier (combiner) plusieurs programmes et les lancer ensemble. Chacun de vos programmes peut faire référence à un symbole (par exemple appeler une fonction) qui est défini dans un autre module objet.

Cette référence est donc dite externe puisqu'elle concerne une entité créée à l'extérieur du module.

MS-LINK combine donc plusieurs modules objets en un seul programme relogeable (qui peut être chargé en une zone quelconque de la mémoire par le système d'exploitation). Ce programme devient alors un fichier exécutable (signalé par l'extension .EXE) ou encore "stand alone" : qui se suffit à lui-même. Au fur et à mesure qu'il lie les modules, MS-LINK s'assure que les références externes sont bien définies quelque part dans un autre module. Si ce n'est pas le cas, il va aller rechercher dans des bibliothèques (fichiers spécialisés contenant des fonctions toutes prêtes) les références externes dont il besoin.

MS-LINK produit aussi un fichier texte qui détaille les références externes résolues et indique les erreurs rencontrées.

MS-LINK utilise autant de mémoire disponible qu'il le peut. Si jamais elle se révélait insuffisante (dans des proportions raisonnables, évidemment), il travaille sur un fichier temporaire nommé VM.TMP.

La figure 10.1 illustre la manière dont s'organise la création d'un fichier exécutable.

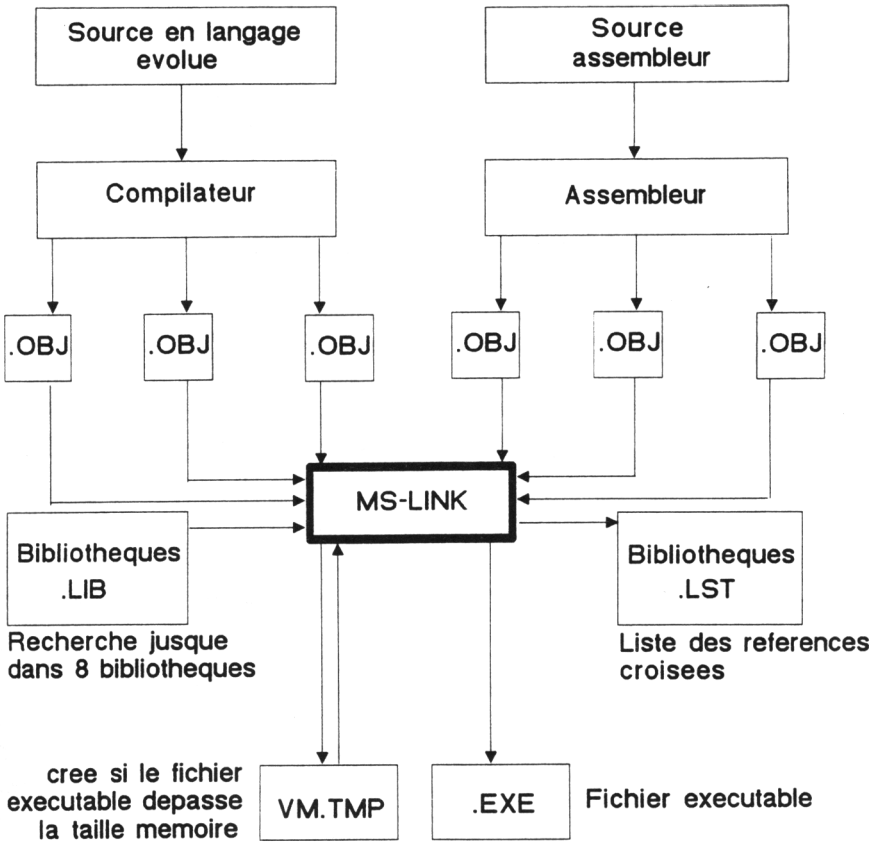


FIGURE 10.1

L'utilisation de MS-LINK

10.3 DEFINITIONS DE BASE

Nous allons expliquer certains des termes figurant dans cette annexe pour vous permettre de comprendre comment MS-LINK fonctionne. Généralement, si vous linkez des modules compilés en BASIC, Pascal ou d'autres langages de haut niveau, la connaissance de ces termes ne vous est pas indispensable. En revanche, si vous écrivez des programmes en assembleur, vous aurez besoin de maîtriser l'utilisation et la terminologie des éditeurs de liens.

Sous MS-DOS, la mémoire peut être divisée en segments, classes et groupes. La figure 10.3 illustre ces concepts.

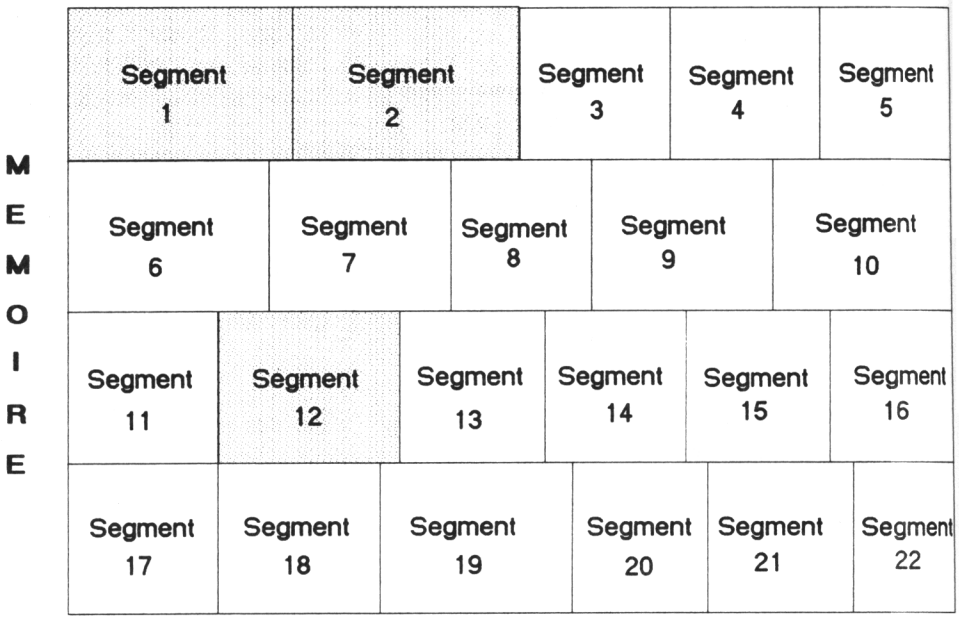


FIGURE 10.3

Division de la mémoire

Exemple :

Numéro de segment	Nom	Classe
1	PROG.1	CODE
2	PROG.2	CODE
12	PROG.3	DONNEES

Notez que les segments 1, 2 et 12 portent des noms différents et peuvent ou non appartenir à la même classe. Les segments 1, 2 et 12 forment un groupe ayant comme adresse de groupe l'adresse la plus basse du segment 1 (c'est-à-dire la plus basse adresse en mémoire).

Chaque segment porte un nom de segment et un nom de classe depuis le premier segment rencontré jusqu'au dernier. Tous les segments appartenant à la même classe sont chargés en mémoire de façon contiguë.

Durant son travail, MS-LINK référence les segments par leur adresse en mémoire. Il le fait en trouvant les groupes de segments.

Un groupe est un ensemble de segments pouvant tenir dans un espace mémoire de 64 K. Les segments n'ont pas besoin d'être contigus pour former un groupe (voir l'illustration). L'adresse d'un groupe quelconque est la plus basse adresse d'un segment parmi ceux du groupe. Durant la phase de génération des liens, MS-LINK analyse les groupes puis référence les segments par leur adresse. Un programme peut se composer d'un ou de plusieurs groupes.

Lorsque vous écrivez un programme en assembleur, vous pouvez assigner un groupe et une classe à un segment donné. Avec les langages de haut niveau (BASIC, COBOL, FORTRAN, C, PASCAL, etc...) c'est le compilateur qui s'en charge.

10.4 FICHIERS UTILISES PAR MS-LINK

MS-LINK :

1. Travaille sur un ou plusieurs fichiers d'entrée.
2. Produit deux fichiers en sortie.
3. Peut créer un fichier temporaire.
4. Peut recevoir l'ordre d'effectuer ses recherches de références dans 8 bibliothèques au plus.

Pour chaque type de fichier, l'utilisateur peut donner un spécificateur de fichier en trois parties. Le format de spécification de fichiers avec MS-LINK est le même qu'avec tout programme MS-DOS :

{d:}<nom de fichier>{.<extension>}

où :

d: désigne l'unité de disque. Elle peut aller de A: à O:. Le signe deux-points est toujours obligatoire et fait partie de la désignation de l'unité.

nom de fichier
est un nom de fichier soumis à la syntaxe MS-DOS et d'une longueur de 8 caractères au plus.

.extension
est un groupe de 1 à 3 caractères ajouté au nom (elle en fait même partie intégrante). Le point est indispensable à la désignation d'une extension. Il n'est pas compris dans le nombre maximum de caractères.

10.4.1 Les extensions des fichiers traités

Si les spécificateurs de fichiers à lier (fichiers objets) ne portent pas d'extension, les valeurs suivantes seront prises en compte par défaut :

.OBJ : Code objet provenant d'un assemblage ou d'une compilation.

.LIB : Fichier bibliothèque (en anglais LIBrary) créés pour certains produits tels FORTRAN ou C.

10.4.2 Les extensions des fichiers produits

MS-LINK affecte les extensions suivantes aux deux fichiers qu'il produit :

.EXE : Fichier exécutable. Cette extension ne peut être changée.

.MAP : Liste des références. Cette extension peut être modifiée par l'utilisateur.

10.4.3 Les fichiers temporaires VM.TMP

MS-LINK utilise la mémoire disponible pour son édition de liens. Si les fichiers à lier produisent un fichier dont la taille dépasse celle de la mémoire disponible, MS-LINK crée un fichier temporaire VM.TMP sur l'unité de disques par défaut. Si MS-LINK crée VM.TMP, il affiche le message suivant :

VM.TMP has been created
Do not change diskette in drive <d:>

ce qui signifie *"VM.TMP a été créé. Ne changez pas la disquette du lecteur ..."*. Une fois qu'est apparu ce message, vous ne devez pas changer la disquette du lecteur indiqué (celle de l'unité par défaut) jusqu'à la fin de l'édition de lien. Si la disquette est ôtée, le résultat du travail de MS-LINK sera indéterminé et MS-LINK peut afficher le message :

Unexpected end of file on VM.TMP
(Fin de fichier imprévue sur VM.TMP)

Le contenu de VM.TMP sert à créer le fichier désigné par l'invite (la demande) "RUN File :" qui vous sert à affecter un nom à votre fichier exécutable. VM.TMP étant un fichier de travail, il est détruit à la fin de l'édition de lien.

ATTENTION

Ne vous servez jamais du nom VM.TMP pour l'un de vos fichiers. Si jamais un fichier VM.TMP figure dans le répertoire par défaut et que MS-LINK ait besoin d'un fichier temporaire, il va commencer par effacer le vôtre avant de créer le sien. Le contenu du VM.TMP précédent sera donc perdu.

10.5 COMMENT LANCER MS-LINK

MS-LINK aura besoin de deux formes de paramètres. Une commande d'appel pour le lancer et des réponses aux questions (invites) qu'il vous posera si nécessaire. De plus, plusieurs commutateurs peuvent déterminer des options de fonctionnement. Habituellement vous taperez toutes les commandes à MS-LINK au clavier. Optionnellement, vous pouvez créer un fichier (dit fichier de réponses) qui automatisera l'entrée des ordres et des paramètres. Vous pouvez vous servir de caractères de commande dans votre dialogue avec MS-LINK.

Il y a trois manières de piloter MS-LINK. La première consiste à taper les réponses au clavier au fur et à mesure des invites. La deuxième consiste à appeler MS-LINK avec une ligne de commande complète. La troisième fait appel à un fichier de réponses contenant toutes les commandes nécessaires. Il faut bien évidemment créer le fichier puis indiquer à MS-LINK où il se trouve.

En résumé, on peut invoquer MS-LINK sous trois formes :

Méthode 1 : LINK puis répondre aux invites.

Méthode 2 : LINK <noms de fichiers> {/<options>}

Méthode 3 : LINK @<spécificateur de fichier>

10.5.1 Méthode 1 : Invites

Pour lancer MS-LINK selon la méthode 1, tapez :

LINK

MS-LINK se charge. Il affiche ensuite quatre questions une à une. En répondant à ces questions, vous lancez l'exécution d'un tâche spécifique.

A la fin de chaque ligne, vous pouvez ajouter un ou plusieurs commutateurs précédés du signe traditionnel de la division (/).

Vous trouverez ci-dessous un bref aperçu des invites qui seront détaillées dans la section 10.7.

Object Modules [.OBJ]:
(Modules objets)

Entrez le nom des fichiers objets à lier. Ils doivent être séparés les uns des autres par un espace ou par le signe Plus (+). Si le signe plus est le dernier caractère tapé, l'invite sera réaffichée pour que vous puissiez entrer d'autres noms.

La réponse à cette question est obligatoire ; il n'y a pas de valeur par défaut.

Run File [Object-file.EXE]:
(Fichier exécutable [Fichier_objet.EXE])

Donnez le nom à affecter au fichier exécutable. Par défaut (si vous ne précisez aucune réponse), MS-LINK choisira le nom du premier fichier objet et remplacera son extension par .EXE (vous devez vous en tenir à cette extension et ne pas la changer par la suite).

List File [Run-file.MAP]:
(Fichier liste des références [Fichier_exécutable.MAP])

Donnez le nom de fichier dans lequel figureront la liste des références croisées et les messages du linker.

Le nom par défaut est celui du fichier exécutable en changeant son extension pour .MAP (bien que ce ne soit pas obligatoire, il est souhaitable de la garder car il s'agit d'une convention bien établie).

Librairies []:
(Bibliothèques)

Liste des fichiers où doit avoir lieu la recherche des références externes. Les noms sont séparés par des espaces ou le signe Plus (+). Si le signe plus est le dernier caractère tapé, l'invite sera réaffichée pour que vous puissiez entrer d'autres noms. Par défaut, MS-LINK recherchera dans les bibliothèques par défaut des modules (les extensions seront éventuellement changées en .LIB).

10.5.2 Méthode 2 : Ligne de commande

Pour lancer MS-LINK avec la méthode 2, tapez toutes les commandes sur une seule ligne. Les entrées suivant le mot LINK sont les réponses aux invites. Les différentes réponses doivent être séparées par des virgules. Conformez-vous à la syntaxe suivante :

LINK <liste_fi._obj.>,<fi._exéc.>,<fi._liste>,<liste_biblio> /cs

où :

liste_fi._obj
est la liste des fichiers objets à lier. Les noms sont séparés par le signe plus.

fi._exéc.
est le nom du fichier exécutable produit.

fi.liste
est le nom du fichier liste des références croisées.

liste_biblio
est la liste des fichiers bibliothèques dans lesquels doit s'effectuer la recherche de références.

/cs (commutateurs)
désigne les options (donc non obligatoires) affectant le travail de MS-LINK. Ces commutateurs peuvent être placés tout à la fin de la ligne ou à la suite d'une entrée quelconque, juste avant une virgule.

Lorsque vous préférez choisir la réponse par défaut pour l'une de ces entrées, il vous suffit de ne rien mettre du tout (pas d'espace) entre deux virgules.

Exemple :

```
LINK EXEMPLE+TEXTE+TABLE+NB/P/M,,LISTE,COBLIB.LIB
```

Cette commande provoque le chargement de MS-LINK qui charge ensuite les fichiers EXEMPLE.OBJ, TEXTE.OBJ, TABLE.OBJ et NB.OBJ. MS-LINK marque alors une pause (provoquée par l'option /P). Puis, après que vous ayez appuyé sur une touche, il lie les modules objets et produit une liste des symboles totale (option /M pour Map). Le fichier exécutable prendra par défaut le nom EXEMPLE.EXE ; la liste des références sera dans le fichier LISTE.MAP ; les recherches se feront dans la bibliothèque COBLIB.LIB.

10.5.3 Méthode 3 : Fichier de réponses

Pour lancer MS-LINK selon la troisième méthode, tapez :

LINK @<fichier>

où fichier est le nom de fichier complet (avec le chemin) d'un fichier de réponses. Celui-ci contient les réponses aux invites de MS-LINK (décrites pour la méthode 1) et peut aussi contenir des commutateurs d'options. L'extension du nom du fichier de réponses peut être choisie à votre convenance (hormis .EXE et .COM). Cette méthode 3 vous permet d'invoquer MS-LINK à partir du clavier ou d'un fichier batch sans intervention ultérieure de votre part.

Il est bien évident que cela implique la création préalable d'un fichier de réponses qui se compose de plusieurs lignes de texte, chacune étant une réponse à une invite de MS-LINK. Ces réponses doivent se trouver dans le même ordre que si vous les entriez directement au clavier (voir méthode 1). Si cela est nécessaire, une réponse peut occuper plusieurs lignes (par exemple pour la liste des fichiers objets ou des bibliothèques), chacune d'elles (sauf la dernière) se terminant par un signe plus (+).

Servez-vous des commutateurs optionnels exactement comme vous le feriez au clavier.

Quand la session MS-LINK débute, chaque invite s'affiche et les réponses apparaissent comme si vous étiez en train de travailler au clavier. Si votre fichier ne contient pas les réponses à toutes les invites (par exemple, un nom de fichier manquant, un caractère de commande absent, une ligne composée d'un simple retour chariot), MS-LINK affiche l'invite n'ayant pas reçu de réponse et attend que vous interveniez au clavier. Lorsque vous avez tapé une réponse légale, MS-LINK poursuit son édition de liens.

Exemple de fichier de réponses :

```
EXEMPLE TEXTE TABLE NB
/PAUSE/MAP
LISTE
COBLIB.LIB
```

Ce fichier demande à MS-LINK de charger les 4 fichiers objets (EXEMPLE, TEXTE, TABLE, NB), de faire une pause avant d'éditer une liste complète des symboles pour vous permettre de changer de disquette (voir /PAUSE dans la section consacrée aux commutateurs d'options avant de vous servir de cette option). Après que vous ayez appuyé sur une touche, le fichier produit prendra le nom de EXEMPLE.EXE, la liste LISTE.MAP sera générée et MS-LINK se servira de la bibliothèque COBLIB.LIB avec les options par défaut.

10.6 LES CARACTERES DE COMMANDE

MS-LINK permet l'utilisation de trois caractères de commande :

Plus (+)

Le signe Plus (+) sert à séparer les entrées et à étendre une ligne de commande à la ligne d'écran suivante pour la liste des fichiers objets et des bibliothèques (pour séparer les modules objets on peut mettre un espace à la place de +). Donc, pour taper un grand nombre de réponses (chacune peut être très longue), tapez un signe plus suivi de l'appui sur la touche [RETURN] à la fin de la ligne d'écran avant de se poursuivre. Dans ce cas, MS-LINK affiche une nouvelle invite similaire à la précédente pour que vous puissiez entrer vos réponses complémentaires. Quand le nom de tous les modules à lier ou de toutes les bibliothèques a été donné, finissez la ligne par un appui sur [RETURN] sans bien sûr le faire précéder de +.

Exemple :

```
Object Modules [.OBJ]: EXEMPLE TEXTE TABLE NB+[RETURN]
Object Modules [.OBJ]: NUL+FLIPFLOP+BAB+[RETURN]
Object Modules [.OBJ]: CORSAIR[RETURN]
```

Point-virgule (;)

Pour sélectionner les réponses par défaut à toutes les invites restantes, répondez à l'une d'elles (sauf la première : Run File) par un point-virgule (;) immédiatement suivi d'un retour-chariot. Cette option vous permet de gagner du temps et vous évite d'entrer une série de [RETURN].

Une fois que vous avez tapé et validé (par [RETURN]) un point-virgule, vous ne pouvez plus répondre à aucune invite jusqu'à la fin de l'édition de liens. Donc ne vous servez pas du point-virgule pour ne sauter que quelques réponses mais plutôt, dans ce cas, du retour-chariot.

Exemple :

```
Object Modules [.OBJ]: EXEMPLE TEXTE TABLE NB[RETURN]
Run File [EXEMPLE.EXE]:;[RETURN]
```

Aucune autre invite n'apparaîtra plus et MS-LINK adoptera les réponses par défaut (y compris EXEMPLE.MAP pour la liste des références).

[CONTROL-C] :

La combinaison de touches [CONTROL-C] vous permet d'interrompre le travail de MS-LINK à tout moment. S'il vous arrive de fournir une mauvaise réponse à une invite, comme par exemple un mauvais nom de fichier, vous devez appuyer sur [CONTROL-C] pour sortir de MS-LINK. Vous pourrez ensuite le relancer. Si vous avez tapé une mauvaise réponse mais pas encore appuyé sur la touche [RETURN], vous pouvez évidemment modifier votre réponse en vous servant de la touche [backspace] mais vous ne pouvez revenir à la ligne précédente.

10.7 LES INVITES

MS-LINK vous demande de répondre à quatre questions (on dit invite parce qu'il vous invite à entrer une information). Lorsque vous avez tapé votre réponse à une invite et appuyé sur la touche [RETURN], l'invite suivante apparaît. Après le dernière réponse, MS-LINK commence automatiquement l'édition des liens sans que vous ayez à intervenir de nouveau. A la fin de ce travail, il vous ramène au système d'exploitation. S'il a découvert une erreur, il affiche le message correspondant avant de rendre le contrôle de la machine à MS-DOS.

MS-LINK vous demande les noms des fichiers objets (Object), du fichier exécutable (Run), de la liste des références (List) et enfin des bibliothèques (Libraries). Ces questions apparaissent dans l'ordre que nous venons d'évoquer. Les réponses par défaut sont affichées entre crochets ([et]) à la suite de la question

proprement dite. Vous remarquerez que cela n'est pas le cas pour la première invite puisque, naturellement, MS-LINK ne peut connaître de lui-même le nom des fichiers sur lesquels vous voulez travailler.

Object Modules [.OBJ]:

Tapez la liste des modules objets que vous voulez lier. Par défaut, MS-LINK adopte l'extension .OBJ pour ces fichiers. Vous n'avez donc besoin de préciser l'extension que dans le cas où l'un de vos modules en porterait une autre.

Les noms de module doivent être séparés par le signe Plus (+).

Rappelez-vous que MS-LINK affecte une classe aux segments en fonction de leur ordre de chargement. Vous pouvez donc décider de l'ordre de chargement des segments en fonction de leurs interrelations.

Run File [<Nom_du_premier_module.EXE]:

Cette invite vous permet de décider du nom de fichier exécutable qui sera créé par l'édition de liens. Quelle que soit celle que vous donnerez, le fichier recevra l'extension .EXE.

Si vous n'entrez pas de réponse, MS-LINK prendra le nom du premier module objet à lier et changera son extension pour .EXE.

Exemple:

```
Run File [ESSAI.EXE]: B:TEST/P
```

Cette réponse entraîne MS-LINK à créer le fichier exécutable TEST.EXE sur le lecteur B:. De plus (grâce à /P), MS-LINK marquera une pause pour vous permettre d'insérer une nouvelle disquette apte à recevoir TEST.EXE.

Libraries []:

Les réponses valables vont d'un simple retour-chariot à une liste de 8 noms de fichiers. Dans le premier cas, le linker opte pour la bibliothèque par défaut. Les fichiers bibliothèques doivent être créés avec un utilitaire spécial puisqu'ils répondent à des spécifications bien précises. Par défaut, MS-LINK s'attend à ce que les bibliothèques aient un nom de fichier portant l'extension .LIB.

Les noms de bibliothèques doivent être séparés par des espaces ou des signes plus (+).

Pour résoudre les références externes, MS-LINK cherche dans les bibliothèques selon l'ordre donné par la ligne de commande. Lorsqu'il trouve le module-bibliothèque où figure la référence externe, il le traite comme un autre module objet.

Si MS-LINK ne trouve pas la bibliothèque dans le répertoire désigné, il affiche le message :

Cannot find library <nom_de_bibliothèque>. Type new drive letter.

Il vous demande donc de taper le spécificateur d'une autre unité de disques. Par exemple, vous pouvez entrer B (sans les deux-points).

10.8 LES OPTIONS MS-LINK

Les sept commutateurs (signe / suivi d'une lettre ou d'un mot) de MS-LINK vous permettent diverses modifications de son mode de fonctionnement. Les commutateurs peuvent figurer à la fin d'une réponse à une invite quelle que soit la méthode de lancement que vous employez. Vous pouvez aussi les faire figurer à la fin d'un des champs de votre réponse, juste avant la virgule. Ils peuvent être groupés ou répartis. Même dans le cas où vous faites figurer plusieurs commutateurs groupés, chaque lettre ou mot doit être précédé du signe /.

Tous les commutateurs peuvent être entrés sous forme abrégée mais si vous faites figurer plusieurs lettres, elles doivent être écrites dans l'ordre du mot entier sans en omettre une. Par exemple, pour l'option /DSALLOCATE :

Autorisé	Interdit
/D	/DSL
/DS	/DAL
/DSA	/DLC
/DSALLOCA	/DSALLOCT

/DSALLOCATE

La présence de ce commutateur amène MS-LINK à charger les données à l'extrémité supérieure du segment de données (data). En temps normal, MS-LINK charge les données dans la partie basse du Data Segment. Au moment de l'exécution, le pointeur de DS reçoit la plus basse valeur possible qui permette l'adressage de l'intégralité du segment DS. L'utilisation de cette option en conjonction avec un chargement bas (c'est-à-dire sans se servir de l'option /HIGH ; voir cette section) permet à l'application d'allouer dynamiquement de la mémoire en dessous de la zone spécifiquement allouée pour le groupe D (DGroup) tout en restant adressable avec le pointeur de DS. Les programmes Pascal et FORTRAN ont besoin d'une telle allocation dynamique.

Note : Votre application peut faire une allocation dynamique allant jusqu'à 64 K (s'il reste suffisamment de mémoire) moins le bloc alloué au DGroup.

/HIGH

L'emploi de l'option /HIGH force MS-LINK à placer le programme aussi haut que possible en mémoire. Sinon, MS-LINK (et par conséquent MS-DOS lors de l'exécution) place le programme en bas de la mémoire.

Important : Ne vous servez pas de /HIGH avec des fichiers PASCAL ou FORTRAN.

/LINENUMBERS

Cette option indique à MS-LINK d'inclure les numéros de ligne et les adresses correspondant aux sources des modules objets dans la liste des références. Par défaut, les numéros de ligne ne sont pas rapportés.

Note : Tous les compilateurs ne fournissent pas des modules objets permettant de demander cette option. Dans ce cas, évidemment, les numéros de lignes seront absents.

/MAP

/MAP amène MS-LINK à lister tous les symboles publics (global) définis dans les modules d'entrée. En l'absence de cette option, MS-LINK ne fournira que la liste des erreurs (y compris les références globales indéfinies).

Les symboles sont classés par ordre alphabétique. Pour chacun d'eux, MS-LINK donne sa valeur et son adresse segment:offset dans le fichier exécutable. Les symboles figurent à la fin du fichier Liste.

/PAUSE

L'option /PAUSE demande à MS-LINK de s'arrêter durant la session d'édition de liens au moment de sa rencontre avec le commutateur /PAUSE. Normalement, MS-LINK effectue son travail en continu. /PAUSE permet à l'utilisateur d'échanger les disquettes avant que MS-LINK se mette à écrire le fichier .EXE.

Quand il rencontre /PAUSE, MS-LINK affiche le message suivant :

About to generate .EXE file
Change disks hit any key

(Sur le point de générer le fichier .EXE)
(Echangez les disquettes et frappez une touche)

MS-LINK reprend le traitement quand l'utilisateur frappe une touche.

Attention : N'enlevez pas la disquette devant recevoir le fichier Liste ni celle où (éventuellement) figure VM.TMP.

/STACK:n

n représente un nombre d'octets allant jusqu'à 65535. Ce nombre peut être exprimé dans la base voulue en se conformant aux conventions du langage C (c'est-à-dire que l'on fait précéder un nombre hexadécimal de 0x). Si une valeur comprise entre 1 et 511 est donnée, MS-LINK prendra 512 octets pour la pile. Si vous ne donnez pas de commande /STACK, MS-LINK calculera la taille de la pile automatiquement.

Tous les programmes assembleur ou compilés doivent fournir, dans les modules objet, les informations nécessaires au linker pour qu'il fasse ce calcul.

Un fichier objet au moins doit présenter une instruction d'allocation mémoire pour la pile. Si ce n'est le cas, MS-LINK affichera :

WARNING : NO STACK STATEMENT
(Attention : pas d'instructions quant à la pile)

/NO

/NO est l'abréviation de /NODEFAULTLIBRARYSEARCH (pas de recherche dans la librairie par défaut). Cette option permet d'outrepasser la "tendance" de MS-LINK à faire ses recherches dans les librairies par défaut, si par exemple vous avez ailleurs redéfini des références. Par exemple, si vous liez un module PASCAL, le commutateur /NO ordonne à MS-LINK de ne pas aller rechercher automatiquement les références externes dans PASCAL.LIB.

10.9 EXEMPLE D'EDITION DE LIENS

Cet exemple vous montre le déroulement d'une session de travail avec MS-LINK.

En réponse à l'invite MS-DOS, entrez :

```
LINK
```

Le système affiche les messages suivants (nous faisons aussi figurer les réponses que vous pourriez donner) :

```
MICROSOFT Object Linker V.2.00 (C) Copyright 1982 by Microsoft Inc.

Object Modules [.OBJ]:IOSYSINT [RETURN]
Run file [IO.EXE]:[RETURN]
List File [NUL.MAP]:IO /MAP [RETURN]
Libraries [.LIB]:; [RETURN]
```

Notes :

1. En spécifiant MAP, vous obtenez à la fois un listing alphabétique et chronologique des symboles publics.

2. En répondant PRN à l'invite List File, vous pourriez obtenir un listing imrimé.

3. En ajoutant le commutateur /LINE, MS-LINK vous donne la liste de tous les numéros de lignes de votre module (notez que le fichier risque d'être long).

4. En appuyant sur [RETURN] en réponse à l'invite des bibliothèques, la recherche de référence se fait de manière automatique.

Une fois que MS-LINK a localisé toutes les bibliothèques, il affiche une liste de segments dans le même ordre que celui de leur apparition dans les modules. Cela peut ressembler à ceci :

Start	Stop	Length	Name
00000H	009ECH	09EDH	CODE
09F0H	01166H	0777H	SYSINTSEG

Les colonnes Start et Stop donnent les déplacements sur 20 bits (héxa) de chaque segment par rapport à 0 (0 étant le début du module chargé). Il ne s'agit pas des adresses absolues de chargement. Reportez-vous au manuel du macro assembleur Microsoft pour connaître la manière dont est calculée l'adresse "0" et comment l'on peut en tirer l'adresse absolue d'un segment.

Comme le commutateur /MAP avait été ajouté, la liste des symboles publics classés par noms et valeurs apparaît. Par exemple :

ADDRESS	PUBLICS BY NAME
009F:0012	BUFFERS
009F:005	CURRENT_DOS_LOCATION
009F:001	DEFAULT DRIVE
009F:000B	DEVICE_LIST
009F:0013	FILES
009F:0009	FINAL_DOS_LOCATION
009F:000F	MEMORY_SIZE
009F:0000	SYSINT

ADDRESS	PUBLICS BY VALUE
009F:0000	SYSINT
009F:005	CURRENT_DOS_LOCATION
009F:0009	FINAL_DOS_LOCATION
009F:000B	DEVICE_LIST
009F:000F	MEMORY_SIZE
009F:001	DEFAULT DRIVE
009F:0012	BUFFERS
009F:0013	FILES

10.10 LES MESSAGES D'ERREUR

Toutes les erreurs entraînent l'abandon de l'édition de liens. Après que le problème ait été localisé et qu'on y ait porté remède, il faut relancer MS-LINK.

Attempt to access data outside of segments bounds, possibly bad object module. (Tentative d'accéder à des données en dehors de limites d'un segment, possibilité de module objet défectueux).

Un des fichiers objets a probablement un problème.

Bad numeric parameter. (Mauvais paramètre numérique)

Une expression numérique comprend une ou des lettres.

Cannot open temporary file. (Ouverture du fichier temporaire impossible)

MS-LINK ne peut créer VM.TMP car le disque est plein. Insérez-en un nouveau.

Error : dup record too complex (Enregistrement DUP trop complexe)

Un DUP de votre source assembleur est trop complexe. Essayez de le décomposer.

Error : fixup offset exceeds field width (l'offset exède les limites du champ)

Une ligne assembleur se réfère à une adresse avec une instruction short au lieu d'un long. Corrigez et recommencez l'assemblage.

Input file read error (Erreur de lecture d'un fichier à traiter)

Un des fichiers objets est endommagé.

Invalid object module (Module objet invalide)

Un fichier objet est mal formé ou incomplet (un assemblage interrompu). Ou encore, il ne s'agit pas d'un fichier objet.

Symbol defined more than once. (Symbole défini plus d'une fois)

MS-LINK a trouvé deux (ou plus) définitions du même symbole ou deux définitions portant le même nom.

Program size or number of segments exceeds capacity of linker (La taille du programme ou le nombre de segments dépasse la capacité du linker)

La taille totale ne doit pas dépasser 384 K et le nombre de segments est limité à 255.

Request stack size exceeds 64k (la taille demandée pour la pile dépasse 64 K).

Pour dépasser cette limite de 64 K, servez-vous de /STACK.

Segment size exceeds 64k (la taille d'un segment dépasse 64 K)

64 K constituent la limite du système.

Symbol table capacity exceeded (Capacité de la table des symboles dépassée)

Votre programme fait appel à de nombreux noms ou ceux-ci sont très longs. La table a une taille d'environ 25 K.

Too many external symbols in one module (trop de symboles externes dans un module)

La limite est de 256 par module.

Too many groups (Trop de groupes)

La limite est de 10 groupes.

Too many libraries specified (Trop de bibliothèques demandées)

La limite est de 8 bibliothèques.

Too many public symbols (Trop de symboles publics)

La limite est de 1024 symboles publics.

Too many segments or classes (Trop de segments ou de classes)

La limite est de 256 (classes et segments additionnés)

Unresolved externals :<liste> (Références externes non résolues)

Il manque des définitions dans des modules et les bibliothèques. (Il arrive souvent que certains symboles inconnus soient dus à des fautes de frappe).

VM read error (Erreur de lecture sur VM.TMP)

Erreur disque. Elle ne provient pas de MS-LINK.

Warning : no stack segment (Attention pas de segment de pile)

Il faut qu'un des modules contiennent un allocation de pile. Ce n'est pas le cas mais vous avez donné une instruction /STACK.

Warning : segment of absolute or unknown type (Attention : segment de type absolu ou inconnu)

Un module est défectueux ou l'on a tenté de lier un module que MS-LINK ne peut traiter (par exemple un module absolu).

Write error in tmp file (Erreur d'écriture dans un fichier temporaire)

Il n'y a plus assez de place pour allonger VM.TMP.

Write error on run file (Erreur d'écriture du fichier exécutable)

Habituellement, il ne reste plus assez de place pour écrire le fichier exécutable.

ANNEXE 11

L'INTERPRETEUR DE COMMANDES COMMAND.COM

Lorsque MS-DOS se charge, il appelle à son tour COMMAND.COM qui est l'interpréteur de commandes résident standard. Vous pouvez cependant appeller l'interpréteur de commandes grâce à l'ordre ci-dessous (en fait il s'agit d'un niveau inférieur de COMMAND.COM) :

command {<lecteur><chemin>}{<canalctty>}{/p}{/c <chaîne>}{/e:<nnnn>}

Cet ordre charge un nouvel interpréteur de commandes, c'est-à-dire un programme MS-DOS permettant à l'utilisateur de gérer le système par des ordres tapés au clavier.

COMMAND.COM se charge en deux parties : une partie transitoire et une partie résidente. Certaines applications peuvent écraser la partie transitoire de COMMAND.COM durant leur exécution. Lorsque cela se produit la partie résidente de COMMAND.COM recharge, quand elle reprend le contrôle de la machine, la partie écrasée à partir du disque.

<lecteur> et *<chemin>* indiquent à COMMAND.COM le répertoire à partir duquel il pourra recharger sa partie transitoire (puisque grâce à config.sys, COMMAND.COM n'est pas forcément dans le répertoire racine).

<canalctty> permet de désigner une autre console que con: comme périphérique d'entrées-sorties par défaut (par exemple un terminal relié à com1:).

/e détermine la taille d'environnement où nnnn est une valeur en octets. Cette taille peut aller de 128 à 32768 octets. La valeur par défaut est 128. Si la valeur fournie est inférieure, MS-DOS affecte tout de même 128 octets mais affiche le message d'erreur : *"Invalid environment size specified"*. Si nnnn est supérieur à 32768, MS-DOS affiche le même message mais affecte 32768 octets.

/p demande à COMMAND.COM de refuser les demandes EXIT vers un niveau supérieur.

Enfin, si le commutateur /c est présent (il doit être le dernier de la ligne), il indique que la chaîne de caractères qu'il précède doit être exécutée. Il faut évidemment que cette chaîne soit cohérente et constitue un ou plusieurs ordres MS-DOS. Au lancement du COMMAND.COM appelé, la chaîne suivant /c sera exécutée par l'interpréteur de commandes qui reprendra ensuite le contrôle. Cette chaîne doit donc se conformer à la même syntaxe que celle que vous auriez tapée au clavier et peut comprendre des commutateurs.

Exemple :

command /c chkdsk b:

Cet exemple demande à COMMAND.COM

1. d'appeler un COMMAND.COM de niveau inférieur,

2. de lancer le programme chkdsk pour vérifier la disquette de B:

3. de revenir au premier interpréteur de commandes.

ANNEXE 12

L'UTILITAIRE DE DEBUGGAGE DEBUG

12.1 INTRODUCTION

Le DEBUG Utility de Microsoft (DEBUG) est un utilitaire de débuggage permettant l'examen et la mise au point des fichiers binaires et exécutables. On pourrait condidérer qu'alors qu'EDLIN est un outil de modification des fichiers sources, DEBUG en est le pendant en ce qui concerne les fichiers binaires. Il permet de se dispenser d'un ré-assemblage d'un programme qui n'avait besoin que d'une modification mineure mais surtout de tester cette modification avant de l'apporter effectivement au fichier source. Il vous permet donc d'altérer le contenu d'un fichier, de changer et d'examiner le contenu des registres du processeur central et de lancer une routine immédiatement afin de s'assurer de la validité de ces changements.

Les commandes DEBUG peuvent être abandonnées à tout moment en appuyant sur [CONTROL-C]. [CONTROL-S] suspend l'affichage de manière à ce que l'on puisse le lire avant un scrolling ; l'appui sur toute autre touche que [CONTROL-C] ou [CONTROL-S] fait repartir l'affichage. Vous remarquez évidemment que la logique de ces commandes est cohérente avec celles du niveau de commande de MS-DOS.

12.2 COMMENT LANCER DEBUG

On peut utiliser DEBUG de deux manières. Selon la première méthode, vous tapez vos commandes en réponse au signe d'invite de DEBUG (un trait d'union). Avec la seconde, vous entrez tous vos ordres sur la ligne de commande appelant DEBUG.

En résumé, cela donne :

Méthode 1 : DEBUG
Méthode 2 : DEBUG {<specifich> {<arguments>}}

12.2.1 Méthode 1 : DEBUG

Avec la première méthode, vous lancez DEBUG en tapant :

DEBUG

l'utilitaire affiche alors son signe d'invite (-) vous indiquant qu'il est prêt à accepter des commandes. Etant donné que l'on n'a pas précisé de nom de fichier, on peut à partir de là accéder à une zone mémoire, un secteur de disque ou à un fichier disque. Il suffira d'entrer la commande appropriée.

Attention :

1. Lorsque l'on invoque DEBUG (version 2.0), il met en place un en-tête de programme à l'offset 0 de la zone de travail du programme. Avec les précédentes versions de DEBUG, vous pouviez écraser cet en-tête. Vous pouvez toujours venir écrire dans cette zone d'en-tête si vous n'avez pas précisé de spécificateur de fichier à DEBUG. Mais, si vous débuggez un fichier .COM ou .EXE, vous ne devez pas toucher à cet en-tête en dessous de l'adresse 5CH ou DEBUG se terminera de manière peu élégante.

2. Ne relancez pas un programme après que le message *"Program terminated normally"* ait été affiché. Vous devez le recharger avec les commandes N et L car vous ne pouvez savoir s'il a ou non modifié son propre code (au sens large : données et allocations comprises).

12.2.2 Méthode 2 : Ligne de commande

Pour exécuter DEBUG selon la seconde méthode, tapez :

DEBUG {<specifich> {<arguments>}}

Par exemple, avec un spécificateur de fichier dans cette ligne :

DEBUG FICHIER.EXE

DEBUG charge alors le fichier FICHIER.EXE depuis l'adresse 100 (hexa) du segment disponible le plus bas. Les registres BX:CX reçoivent le nombre d'octets chargés en mémoire.

Dans le cas de la présence d'un spécificateur de fichier, on peut ajouter une liste d'arguments. Il s'agit des paramètres (par exemple des noms de fichiers ou des commutateurs) que l'on passerait en temps normal au programme à débugger. On obtient donc l'équivalent en mémoire d'une commande du type :

<nom_de_programme> <paramètre(s)_du_programme>

12.3 LES COMMANDES DEBUG

Chaque commande DEBUG se compose d'une lettre unique suivie d'un ou de plusieurs paramètres. De plus, on peut utiliser les caractères de contrôle et les fonctions d'édition spéciales (F1, F3 par exemple) décrites dans le guide utilisateur de MS-DOS.

Si vous commettez une erreur de syntaxe, DEBUG ré-affiche la ligne de commande et indique la position de l'erreur par une flèche vers le haut (^) suivie du mot "erreur".

Par exemple :

```
dcs:100 cs:110
        ^erreur
```

Les paramètres de commande peuvent être tapés aussi bien en majuscules qu'en minuscules.

Le tableau 12.1 résume les commandes de DEBUG. Celles-ci seront décrites en détail par la suite avec leurs paramètres.

Tableau 12.1 : les commandes DEBUG

Commande	Fonction
A{<adresse>}	Assemble
C<zone><adresse>	Compare
D{<zone>}	Affiche (Dump) mémoire
E<adresse>{<liste>}	Affecte (Enter) une (des) valeur(s)
F<zone><liste>	Remplit (Fill)
G{=<adresse>{<adresse>...}}	Exécute (Go)
H<valeur><valeur>	Hexa
I<port>	Entre (Input) un valeur d'un port
L{<adresse>{<lecteur><enreg.><enreg.>}}	Charge (Load)
M<zone><adresse>	Déplace (Move)
N<nom_fich.>{<nom_fich.>}	Nomme
O<port><octet>	Envoie (Output)
Q	Quitte
R{<registre>}	Registre
S<zone><liste>	Cherche (Search)
T{=<adresse>}{<valeur>}	Trace
U{<zone>}	Désassemble (Unassemble)
W{<adresse>{<lecteur><enreg.><enreg.>}}	Ecrit (Write)

12.4 PARAMETRES

Toutes les commandes DEBUG (sauf Quit) acceptent des paramètres. Ils peuvent être séparés par des délimiteurs (espace ou virgule) qui ne sont obligatoires qu'entre deux valeurs hexadécimales consécutives. Par conséquent, les commandes suivantes sont équivalentes :

dcs:100 110 d cs:100 110 d,cs:100,110

et permettent l'affichage mémoire de 100h à 110h.

PARAMETRE	DEFINITION
<lecteur>	Valeur hexadécimale à un chiffre indiquant le lecteur concerné par l'opération de chargement ou d'écriture d'un fichier. Les valeurs autorisées vont de 0 à 3. Elles désignent les unités de disques comme ceci : 0=A, 1=B, 2=C, 3=D.
<octet>	Valeur hexadécimale à deux chiffres à affecter où à lire dans une adresse ou un registre.
<enreg.>	(Enregistrement). Une valeur hexadécimale de 1 à 3 chiffres désignant le numéro d'enregistrement logique sur une unité de disque et le nombre de secteurs faisant l'objet d'une opération de lecture ou d'écriture. Un enregistrement logique correspond à un secteur mais leur numérotation diffère dès qu'ils représentent l'intégralité de l'espace du disque.
<valeur>	Valeur hexadécimale d'au plus quatre chiffres servant à spécifier le nombre de fois qu'une commande doit être répétée.
<adresse>	Désignation d'un emplacement mémoire en deux parties : la première peut être soit la désignation alphabétique d'un registre de segment soit une

valeur (4 chiffres hexa), la seconde est un offset (déplacement) en 4 chiffres hexadécimaux. L'expression de segment peut être omise dans le cas où l'on désire adresser le segment par défaut. Il s'agit de DS pour toutes les fonctions sauf G, L, T, U et W pour lesquelles le segment par défaut est CS.

Par exemple : CS:0100
04BA:0100

Le signe deux-points (:) est obligatoire entre la désignation du segment et l'offset.

<zone> Exprimée par deux adresses (par exemple <adresse>, <adresse>) ou une adresse et (éventuellement) l'expression "L<valeur>". Valeur étant le nombre de lignes concernées par la commande.

Exemple : CS:100 110
CS:100 L 10
CS:100

La formulation suivante est interdite :

```
CS:100 CS:110
       ^erreur
```

Par défaut, DEBUG utilise (pour les fonctions concernées comme la dernière ligne valide de notre exemple) l'expression L80. La limite supérieure de <valeur> est 1000. <valeur> est toujours exprimée en hexadécimal.

<liste> Une série de valeurs d'octets ou une chaîne de caractères. <liste> doit être le dernier paramètre de la commande.

Exemple : fcs:100 42 45 52 54 41

<chaîne> Nombre quelconque de caractères délimités par des apostrophes (') ou des guillemets. Si le délimiteur doit apparaître dans la chaîne, il doit être doublé. Par exemple, les chaînes suivantes sont autorisées :

'Voilà une "chaîne" est légal'
'Voilà une ''chaîne'' est légal'
"Voilà une 'chaîne' est légal"
"Voilà une ""chaîne"" est légal"

Sont interdites :

'Voila une 'chaîne' mauvaise'
"Voila une "chaîne" mauvaise"

Remarquez bien que les délimiteurs qui entourent le mot chaîne sont ici inutiles :

'Voilà une ""chaîne"" est légal mais inutile'
"Voilà une ''chaîne'' est légal mais inutile"

Les valeurs ASCII de ces caractères de la chaîne forment une <liste> d'octets.

NOM :

(A)SSEMBLE

ACTION :

Assemble directement en mémoire les mnémoniques 8086 / 8087 /8088.

SYNTAXE :

A{<adresse>}

S'il découvre une erreur de syntaxe, DEBUG signalera

```
^error
```

et affichera de nouveau l'adresse d'assemblage courante.

Toutes les valeurs numériques seront exprimées en hexadécimal et doivent comprendre de 1 à 4 caractères. Les mnémoniques préfixes doivent apparaître juste après l'opérateur concerné. Une seule instruction par ligne est autorisée.

Les mnémoniques de segments sont CS:, DS:, ES: et SS:. La mnémonique d'un retour long est RETF (far). Les mnémoniques de manipulation de chaînes doivent faire mention explicitement de la longueur de la chaîne. Par exemple, utilisez MOVSW pour déplacer les chaînes sur un mot et MOVSB pour déplacer les chaînes sur un octet.

L'assembleur différenciera automatiquement les appels courts (short), proches (near) ou longs (far) en fonction du déplacement vers l'adresse cible. Vous pouvez cependant outrepasser cet automatisme avec les préfixes NEAR ou FAR. Par exemple :

```
0100:0500      JMP 502          ; saut court de 2 octets
0100:0502      JMP NEAR 505     ; saut proche de 3 octets
0100:505       JMP FAR 50A      ; saut long de 5 octets
```

Le prefixe NEAR peut être abrégé en NE mais le préfixe FAR doit être écrit en entier.

DEBUG ne peut savoir si certaines opérandes se réfèrent à la valeur d'une adresse sur un mot ou sur un octet. Il faut alors l'expliciter avec les préfixes WORD PTR ou BYTE PTR (que l'on peut abréger respectivement en WO et BY). Par exemple :

```
NEG      BYTE PTR [128]
DEC      WO [SI]
```

De même, DEBUG ne peut savoir si une opérande se réfère à une adresse ou s'il s'agit d'une opérande immédiate.

Par convention, les opérandes se référant à une adresse doivent donc être écrites entre crochets. Par exemple :

```
MOV     AX,21       ; Donne la valeur 21h à AX
MOV     AX,[21]     ; Charge AX avec le contenu de l'adresse 21h
```

Assemble accepte deux pseudo-instructions courantes. DB permettra d'écrire des valeurs sur un octet en mémoire et DW permettra d'écrire des valeurs sur un mot. Par exemple :

```
DB      1,2,3,4
DB      'Un message',0
DB      "Une apostrophe :'"
DW      1000,2000,3000
```

Assemble permet l'utilisation d'expressions composées sur les registres :

```
ADD     BX,34[BP+2].[SI-1]
PUSH    [SI]
```

Tous les synonymes des opérateurs sont acceptés. Par exemple :

```
LOOPZ   100
LOOPE   100

JA      200
JNBE    200
```

En ce qui concerne les opérateurs 8087, WAIT et FWAIT doivent être explicitement indiqués. Par exemple :

```
FWAIT FADD ST,ST(3) ; Cette ligne assemble un préfixe FWAIT
LD TBYTE PTR [BX]   ; Mais pas celle-ci
```

NOM :

(C)OMPARE

ACTION :

Compare une portion de la mémoire spécifiée par <zone> avec une portion de même taille commençant à <adresse>

SYNTAXE :

C<zone> <adresse>

Si les deux zones comparées sont identiques, DEBUG n'affiche aucun message et vous renvoie à son signe d'invite. S'il trouve des différences, il les indique sous cette forme :

<adresse1> <octet1><octet2><adresse2>

Par exemple, les commandes suivantes sont équivalentes :

C100,1FF 300 ou **C100 L100 300**

Elles demandent de comparer la zone de mémoire de 100h à 1FFh avec la zone s'étendant de 300h à 3FFh.

NOM :

(D)UMP

ACTION :

Affiche le contenu de la zone mémoire indiquée.

SYNTAXE :

D{<zone>}

Si une zone de mémoire est spécifiée, son contenu entier s'affiche. Si l'on tape la commande D sans paramètre, DEBUG affiche le contenu de 128 octets à partir de la première adresse après la dernière affichée par la commande Dump précédente.

L'affichage est divisé en deux parties : les valeurs hexadécimales d'un côté et leur équivalent ASCII de l'autre.

Les caractères non affichables y sont remplacés par un point. Chaque ligne montre 16 octets et donc le contenu de 16 adresses avec un trait d'union entre le huitième et le neuvième octet. Chaque ligne d'écran commence à une adresse multiple de 16.

Si vous entrez la commande :

dcs:100 10F

DEBUG affiche le contenu mémoire sous la forme suivante :

```
04BA:0100 54 4F 4D 20 53 41 57 59-45 52 24 00 00 09 4E 44
TOM SAWYER$....ND
```

(La ligne est coupée en deux dans ce manuel pour des raisons typographiques).

Si vous tapez la commande :

D

l'affichage a la forme décrite auparavant. Chaque ligne commence par une adresse égale à l'adresse de la ligne précédente plus 16. Chaque commande D affichera ensuite les adresses suivant la dernière affichée.

Si vous demandez :

DCS:100 115

vous obtenez le contenu des adresses 100h à 115h du segment CS.

NOM :

(E)NTER

ACTION :

Affecte les valeurs spécifiées sur 8 bits à partir de l'adresse spécifiée.

SYNTAXE :

E<adresse> {<liste>}

La liste peut ne comprendre qu'une seule valeur (un octet) ou plusieurs et dans ce dernier cas l'incrémentation des adresses est séquentielle et automatique. Si une erreur est détectée, aucune valeur n'est affectée.

Si vous entrez <adresse> sans second paramètre, DEBUG affiche cette adresse là avec son contenu, puis répète l'adresse sur la ligne suivante et attend que vous interveniez au clavier. Vous pouvez donc :

1. Remplacer le contenu sur un octet avec la valeur que vous tapez. Si cette valeur hexadécimale n'est pas correcte ou si elle comprend plus de deux chiffres, le caractère illégal ou le chiffre en surnombre n'est pas affiché.

2. Appuyer sur la barre d'espacement pour passer à l'octet suivant. Pour changer la valeur de cette adresse, vous pouvez alors procéder comme décrit en 1. Si vous appuyez plus de 7 fois encore sur la barre d'espacement, DEBUG recommence à afficher les octets sur la ligne suivante en faisant commencer celle-ci par l'adresse de son premier octet.

3. Taper un trait d'union pour revenir à l'octet précédent. Lorsque vous tapez ce signe, DEBUG affiche une nouvelle ligne avec l'adresse au début et les valeurs sur un octet.

4. Appuyer sur la touche [RETURN] pour sortir de la commande Enter. Vous pouvez faire cela quelle que soit la position courante.

Par exemple, supposons que nous entrions la commande suivante :

ECS:100

DEBUG affiche :

04BA:0100 EB.

Pour changer cette valeur en 41, tapez 41 :

04BA:0100 EB.41

Pour aller vers les octets suivants, appuyez sur la barre d'espacement :

04BA:0100 EB.41 10. 00. BC.

Pour changer le BC en 42 :

04BA:0100 EB.41 10. 00. BC.42

Puis vous décidez que le 10 devrait plutôt être 6F ; vous tapez le signe moins jusqu'à revenir à l'octet 01 (valeur 10) et le remplacez par 6F :

04BA:0100 EB.41 10. 00. BC.42-
04BA:0102 00.-
04BA:0101 10.6F

L'appui sur la touche [RETURN] vous permet de revenir au niveau de commande de DEBUG.

NOM :

(F)ILL

ACTION :

Affecte aux adresses de <zone> les valeurs de <liste>.

Si la zone comprend plus d'octets que le nombre de valeurs dans la liste, la liste sera répétée pour remplir toute la zone spécifiée. Si la liste contient plus de valeurs que le nombre d'octets de la zone, les valeurs en trop seront ignorées.

Si un emplacement mémoire de la zone n'est pas accessible (mémoire défectueuse ou inexistante), l'erreur se répétera pour toutes les adresses suivantes.

Par exemple, avec la commande suivante :

F04BA:100L100 100 57 4B 57 42 41

DEBUG affectera les valeurs spécifiées à la zone allant de 04BA:100 à 04BA:1FF. Les 5 octets indiqués seront répétés jusqu'à remplir (Fill) les 160 adresses de la zone.

NOM :

(G)O

ACTION :

Exécute le programme en mémoire.

SYNTAXE :

G{=<adresse>{<adresse>...}}

Si la commande G est tapée seule, le programme s'exécute comme il le ferait sans DEBUG.

Si vous précisez une adresse, l'exécution commence à cet endroit là. Le signe égal (=) est obligatoire, il permet à DEBUG de reconnaître l'adresse de départ (=<adresse>) d'un point d'arrêt.

Les autres adresses figurant dans la commande permettent de suspendre l'exécution à la première d'entre elles que le compteur ordinal va rencontrer quelle que soit la position de cette adresse (un point d'arrêt en fait) dans la liste. Lorsque l'exécution rencontre un point d'arrêt, DEBUG affiche les registres, les indicateurs (flags) et le désassemblage de la dernière instruction exécutée.

Vous pouvez préciser jusqu'à 10 points d'arrêt. Ils doivent être placés sur le premier octet d'une instruction 8086. Si vous demandez plus de 10 points d'arrêt, DEBUG affiche le message BP error (Erreur de breakpoint).

La pile utilisateur doit être valide et disposer de 6 octets pour cette commande. La fonction G utilise une instruction IRET pour effectuer un saut vers le programme à tester. Le pointeur de pile utilisateur est validé et les flags, CS et IP sont empilés (ce qui implique que si la pile utilisateur n'est pas valide ou qu'il n'y reste plus assez de place, la machine puisse "planter"). Un code d'interruption (0CCh) est placé aux adresses de point d'arrêt spécifiées.

A la rencontre d'un point d'arrêt, tous les points d'arrêt reprennent leur valeur initiale. Si l'exécution n'est pas suspendue par un point d'arrêt, les codes d'interruption ne sont pas remplacés par les instructions initiales.

Par exemple, avec la commande suivante :

GCS:7550

Le programme actuellement en mémoire s'exécute jusqu'à l'adresse 7550 du segment CS. DEBUG affiche alors tous les flags, les registres et la commande Go est terminée.

Si vous relancez Go après la rencontre d'un point d'arrêt, le programme s'exécute comme si vous l'aviez lancé depuis le niveau de commande de MS-DOS. La seule différence est que l'exécution commence à l'instruction suivante le point d'arrêt et non au début du programme.

NOM :

(H)EX

FONCTION :

Affiche la somme et la différence de deux valeurs hexadécimales.

SYNTAXE :

H<valeur><valeur>

DEBUG affiche la somme et la différence des deux valeurs hexadécimales données en paramètres.

Par exemple, vous tapez :

H19F 10A

DEBUG fait les calculs et affiche le résultat :

02A9 0095

NOM :

(I)NPUT

ACTION :

Lit un octet depuis le port spécifié et affiche sa valeur.

SYNTAXE :

I<port>

<port> est une adresse d'E/S sur 16 bits.

Par exemple, avec la commande suivante :

I2F8

DEBUG affiche la valeur :

28

correspondant au contenu (hexa) de l'adresse du port.

NOM :

(L)OAD

ACTION :

Charge un fichier en mémoire.

SYNTAXE :

L{<adresse>{<lecteur><enreg.><enreg.>}}

Le fichier doit avoir été nommé soit à l'appel de DEBUG soit avec la commande N. Dans les deux cas, le nom de fichier est formaté de la manière habituelle pour un FCB. Celui du fichier se trouve en CS:5C.

Si la commande L est entrée sans paramètre, DEBUG charge le fichier à partir de l'adresse CS:100 et donne à BX:CX le nombre d'octets lus. Si une adresse figure dans la commande, c'est à partir de celle-ci que le fichier viendra s'implanter. Si l'on donne tous les paramètres à L, ce sont des secteurs que charge DEBUG et non un fichier. Les enregistrements sont pris sur le lecteur spécifié (on désigne l'unité A: par 0, B: par 1, C: par 2, etc...). Les numéros de secteurs sont les numéros absolus sur le disque. DEBUG commence à charger le premier secteur et continue jusqu'à atteindre le nombre de secteurs indiqué par le deuxième paramètre <enreg.>.

Exemple :

```
A>DEBUG
-NFICHIER.COM
```

nous pouvons charger le fichier nommé par la commande :

L

DEBUG charge le fichier et affiche ensuite son invite habituelle.

Si vous ne voulez charger qu'une portion d'un fichier ou certains secteurs du disque :

L04BA:100 2 0F 6D

DEBUG charge 109 (6D hex) secteurs à partir de l'enregistrement logique 15. Ces données vont s'implanter à partir de l'adresse 04BA:100. Après le chargement, DEBUG vous ramène au signe d'invite.

Dans le cas de fichiers .EXE, l'adresse de chargement est ignorée et le fichier est relogé à l'adresse de chargement figurant dans son en-tête. L'en-tête lui même n'est pas chargé aussi la taille du fichier diffère-t-elle en mémoire de ce qu'elle est sur disque.

Si l'on charge un fichier portant l'extension .HEX sans préciser de paramètre d'adresse, DEBUG le charge à partir de l'adresse figurant dans le fichier.

Si la commande L inclut un paramètre d'adresse, le fichier s'implantera à la somme de l'adresse spécifiée dans la commande et de l'adresse figurant dans le fichier .HEX.

NOM :

(M)OVE

ACTION :

Copie le contenu de la zone mémoire spécifiée à l'adresse indiquée.

SYNTAXE :

M<zone><adresse>

DEBUG effectue une copie intelligente, c'est-à-dire que la zone servant de modèle à la copie et la zone destination peuvent se recouper sans que cela entraîne une perte de données. Les adresses qui pourraient être écrasées sont copiées en premier. Lorsqu'il doit déplacer des données d'une zone d'adresses hautes vers une

zone d'adresses basses, la copie commence depuis le bas du bloc et se poursuit en "remontant". Lorsque la copie doit se faire depuis des adresses basses vers des adresses hautes, la copie des données hautes se fait en premier et se poursuit vers le bas.

Notez que toutes les adresses dont le contenu est "déplacé" ne sont en fait pas affectées et ne reçoivent aucune valeur de remplacement. MOVE effectue bien une copie malgré son nom. C'est pourquoi l'ordre dans lequel MOVE effectue ce travail est important.

Exemple :

MCS:100 110 CS:500

Dans un premier temps, MOVE copie l'adresse CS:110 vers CS:510, puis CS:10F vers CS:50F, etc jusqu'à ce qu'il ait copié CS:100 sur CS:500. Vous pouvez visualiser le résultat de la copie avec la fonction D<adresse>.

NOM :

(N)AME

ACTION :

Spécifie les noms de fichiers.

SYNTAXE :

N<nom_de_fichier> {<nom_de_fichier>...}

La commande NAME assure deux fonctions. Premièrement, NAME permet d'assigner un nom de fichier aux opérations LOAD et WRITE. Par conséquent, si vous lancez DEBUG sans nom de fichier à traiter, vous devrez donner celui-ci avant de pouvoir le charger. Deuxièmement, NAME permet de passer des paramètres au fichier à débugger. Ces paramètres sont identiques à ceux que l'on passerait par la ligne de commande en utilisation normale du programme (fichier).

Ces deux fonctions se confondent en fait. Considérez l'exemple suivant :

```
-NFICHIER.EXE
-L
-G
```

1. NAME désigne le fichier FICHIER.EXE comme étant celui sur lequel agiront les commandes Load et Write.

2. NAME assigne aussi FICHIER.EXE comme étant le premier paramètre utilisé par tout programme qui sera ensuite débuggé.

3. Load charge le fichier FICHIER.EXE en mémoire.

4. Go lance l'exécution de FICHIER.EXE avec comme seul paramètre son propre nom de fichier ce qui reviendrait (si l'on n'était pas sous DEBUG) à taper FICHIER.EXE au niveau de commande MS-DOS.

Une chaîne de commandes plus utile pourrait être :

```
-NFICHIER1.EXE
-L
-NFICHIER2.DAT FICHIER3.DAT
-G
```

Ici, Name détermine que le prochain Load se fera à partir du fichier FICHIER1.EXE. La commande LOAD charge le fichier FICHIER1.EXE, puis on utilise une seconde fois NAME pour spécifier les paramètres nécessaires au lancement de FICHIER1.EXE. Finalement, on appelle la commande GO qui fait s'exécuter FICHIER1.EXE comme si l'on avait tapé FICHIER1 FICHIER2.DAT FICHIER3.DAT au niveau de commande MS-DOS. Notez bien que si l'on faisait encore appel à la fonction Name à ce moment, le fichier FICHIER1.EXE serait sauvegardé sous le nom FICHIER2.DAT défini par la dernière commande Name si nous demandions WRITE. Pour éviter de tels problèmes, il faut toujours exécuter un NAME avant un LOAD ou un WRITE.

Quatre régions de la mémoire peuvent être affectées par la commande Name :

CS:5C	Le FCB du fichier 1.
CS:6C	Le FCB du fichier 2.
CS:80	Comptage des caractères.
CS:81	Tous les caractères tapés.

Lorsque l'on donne un premier paramètre à la commande NAME, cela installe un FCB à CS:5C. Pour un second paramètre, un FCB est créé à CS:6C. Le nombre de caractères tapés dans la commande Name (hormis le N) se trouve à l'adresse CS:80. La chaîne de commande passée par N se trouve elle-même en CS:81 (toujours sans la première lettre N). Cette chaîne peut contenir tous les caractères normalement admis par COMMAND.COM et les commutateurs d'options que l'on fait passer aux programmes en temps normal.

Utilisation typique de la commande Name :

```
DEBUG PROG.COM
-NPARAM1 PARAM2/C
-G
-
```

Dans ce cas la commande Go exécute le fichier PROG.COM de la même manière que si l'on avait tapé sous MS-DOS :

```
PROG PARAM1 PARAM2/C
```

le débuggage se fait donc dans un environnement normal pour PROG.COM.

NOM :

(O)UTPUT

ACTION :

Envoie l'octet spécifié vers le port dont on donne l'adresse.

SYNTAXE :

O<port> <octet>

L'adresse est exprimée sur 16 bits.

Par exemple :

O2F8 4F

envoie la valeur 4F au port dont l'adresse d'E/S est en 2F8.

NOM :

(Q)UIT

ACTION :

Met fin au travail sous DEBUG.

SYNTAXE :

Q

La commande Q n'exige pas de paramètres et fait sortir de DEBUG sans sauvegarde du fichier présent en mémoire. Cette commande ramène donc au niveau de commande de MS-DOS.

Par exemple, pour terminer une session de travail, tapez :

Q[RETURN]

Vous retrouvez le signe d'invite de MS-DOS.

NOM :

(R)EGISTER

ACTION :

Affiche le contenu d'un ou de plusieurs registres du processeur.

SYNTAXE :

R{<registre>}

<registre> représente la mnémonique du registre voulu. S'il est absent, DEBUG affiche le contenu de tous les registres, les flags et l'instruction pointée par le compteur ordinal désassemblée.

Si l'on donne un nom de registre, la valeur sur 16 bits qu'il contient est affichée en hexadécimal et un signe d'invite apparaît. Vous pouvez alors soit taper une valeur à affecter au registre, soit appuyer sur [RETURN] si vous ne voulez pas le modifier.

Les noms de registres autorisés sont :

AX	**BP**	**SS**	
BX	**SI**	**CS**	
CX	**DI**	**IP**	(IP et PC désignent tous deux le
DX	**DS**	**PC**	compteur ordinal)
SP	**ES**	**F**	

Toute autre valeur donnée comme nom de registre entraîne l'apparition du message d'erreur BR.

Si l'on donne F comme nom de registre, DEBUG affiche chaque indicateur (flag) sous la forme d'un code de deux lettres. Vous pouvez configurer le flag comme vous le voulez en entrant son code d'état (voir tableau page suivante).

Flag	Code si mis à 1	Code si mis à 0
Overflow	OV	NV
Direction	DN decrémente	UP incrémente
Interrupt	EI autorisées	DI inhibées
Sign	NG négatif	PL positif
Zero	ZR	NZ
Auxiliary Carry	AC	NA
Parity	PE paire	PO impaire
Carry	CY	NC

Lorsque vous tapez RF, les flags s'affichent sur une ligne dans l'ordre indiqué ci-dessus. A la fin de la liste des flags, DEBUG affiche le signe - (trait d'union). Vous pouvez entrer de nouvelles valeurs pour les drapeaux. Ces nouvelles valeurs peuvent être tapées dans n'importe quel ordre et vous n'avez pas besoin de laisser d'espace entre chaque code de deux lettres. Pour sortir de la commande R, appuyez sur la touche RETURN. Les flags n'ayant pas reçu de nouvelles valeurs restent inchangés.

Si l'on donne plus d'une valeur pour un flag, DEBUG retourne le message *"DF error"*. Si vous donnez un code qui n'existe pas, DEBUG affiche le message *"BF error"*. Dans les deux cas, seuls seront changés les flags modifiés avant l'erreur.

A la mise en route, les registres de segments sont positionnés sur le début (le bas) de la mémoire libre ; le *Instruction Pointer* (PC) est mis à 100h, tous les flags sont effacés et les autres registres mis à 0.

Exemple :

R

Amène DEBUG à afficher tous les registres, flags et l'instruction désassemblée pointée par l'adresse courante. Si l'adresse était CS:11A, vous pourriez obtenir :

```
AX=0E00  BX=00FF  CX=0007  DX=01FF  SP=039D  BP=0000
SI=005C  DI=0000  DS=04BA  ES=04BA  SS=04BA  CS=04BA
IP=011A  NV UP DI NG NZ AC PE NC
04BA:011A   CD21    INT 21
```

En tapant RF, vous obtenez :

NV UP DI NG NZ AC PE NC-

Vous pouvez taper des codes d'indicateurs valides pour modifier le registre d'état :

NV UP DI NG NZ AC PE NC-PLEICY [RETURN]

DEBUG vous renvoie à son signe d'invite. Pour voir le résultat de vos changements, vous pouvez taper R ou RF :

RF
NV UP EI PL NZ AC PE CY-

Appuyez sur la touche [RETURN] pour laisser ces flags intacts, ou changez de nouveau leur valeur.

NOM :

(S)EARCH

ACTION :

Parcourt la zone spécifiée pour y rechercher les valeurs spécifiées.

SYNTAXE :

S<zone><liste>

La liste peut renfermer un ou plusieurs octets séparés par une virgule ou un espace. Si <liste> contient plus d'un octet, seule la première adresse du premier d'entre eux sera retournée. Si la liste ne contient qu'un seul octet, toutes ses occurrences dans la zone seront retournées.

Exemple :

SCS:100 110 41

DEBUG affiche un réponse du type :

04BA:0104
04BA:010D
-

NOM :

(T)RACE

ACTION :

Exécute une instruction et affiche le contenu de tous les registres et flags avec le désassemblage de l'instruction.

SYNTAXE :

T{=<adresse>}{<valeur>}

Si l'adresse optionnelle est précisée, l'exécution en mode trace commence à partir de celle-ci. La <valeur> optionnelle indique à DEBUG d'exécuter en mode trace le nombre de pas correspondant à la valeur.

La commande T fait appel au mode Trace hardware du microprocesseur 8086 (ou 8088).

Par conséquent, vous pouvez aussi "tracer" les instructions en ROM.

Par exemple, si vous tapez :

T

DEBUG retourne la valeur de tous les registres, flags et l'instruction pointée par IP désassemblée. Supposons que la position courante soit 04BA:011A, DEBUG pourrait afficher :

```
AX=0E00  BX=00FF  CX=0007  DX=01FF  SP=039D  BP=0000
SI=005C  DI=0000  DS=04BA  ES=04BA  SS=04BA  CS=04BA
IP=011A  NV UP DI NG NZ AC PE NC
04BA:011A   CD21    INT 21
```

Si vous tapez :

T=011A 10

DEBUG exécute seize (10h) instructions à partir de 011A du segment courant en affichant les registres, flags et les instructions exécutées. Cet affichage défile jusqu'à la dernière instruction demandée. Si vous voulez lire ces valeurs en cours de route (puisque ce n'est qu'à la dernière instruction que l'affichage est suspendu), vous pouvez taper [Ctrl][S] ce qui fige l'écran jusqu'à la pression d'une touche.

NOM :

(U)NASSEMBLE

ACTION :

Désassemble la zone spécifiée et affiche la représentation mnémonique du code avec les adresses et leurs valeurs.

SYNTAXE :

U<zone>

L'affichage d'un code désassemblé ressemble à la source d'un programme assembleur. Si vous entrez la commande U sans paramètre, DEBUG affiche la représentation symbolique de 20h octets à partir de la position courante. Si vous tapez U<zone>, DEBUG désassemble tous les octets de cette zone. Si la zone que vous donnez se résume à une seule adresse, DEBUG désassemble 20 (hexa) octets à partir de celle-ci.

Par exemple :

U04BA:100L10

DEBUG désassemble 16 octets à partir de U04BA:100 :

```
04BA:0100  206472      AND     [SI+72],AH
04BA:0103  69          DB      69
04BA:0104  7665        JBE     016B
04BA:0106  207370      AND     [BP+DI+70],DH
04BA:0109  65          DB      65
04BA:010A  63          DB      63
04BA:010B  69          DB      69
04BA:010C  66          DB      66
04BA:010D  69          DB      69
04BA:010E  63          DB      63
04BA:010F  61          DB      61
```

Si vous demandez :

U04ba:0100 0108

l'affichage est le suivant :

```
04BA:0100  206472      AND     [SI+72],AH
04BA:0103  69          DB      69
04BA:0104  7665        JBE     016B
04BA:0106  207370      AND     [BP+DI+70],DH
```

NOM :

(W)RITE

ACTION :

Sauvegarde le programme en cours de mise au point sur le disque.

SYNTAXE :

W{<adresse>{<lecteur><enreg.><enreg.>}}

Si vous entrez la commande W sans paramètre, BX:CX doit déjà avoir la valeur du nombre d'octets à écrire ; le fichier est écrit à partir de CS:100. Si vous ne donnez qu'une adresse comme périphérique à W, le fichier est écrit à partir de cette adresse. Si vous vous êtes servi des commandes G ou T, il faut remettre BX:CX à zéro avant de lancer la commande Write sans paramètre. Notez que dans le cas où l'on a chargé et modifié un fichier, le nom, la longueur et l'adresse de départ sont correctement définis pour la sauvegarde si la longueur n'a pas changé.

Le fichier doit avoir été nommé, soit lors de l'appel de DEBUG, soit par la commande N (voir Name). L'une ou l'autre de ces méthodes attribuent un nom dont le format est utilisable par la routine de sauvegarde. Le FCB pourra être examiné en CS:5C.

Si tous les paramètres figurent dans la commande, l'écriture commence à partir de l'adresse fournie ; le fichier est sauvegardé sur l'unité <lecteur> (la désignation est numérique : 0=A:, 1=B:, 2=C:, etc...) ; DEBUG commence l'écriture au numéro d'enregistrement logique figurant dans le premier paramètre <enreg.> ; il poursuit l'écriture jusqu'à avoir écrit le nombre de secteurs figurant dans le deuxième paramètre <enreg.>.

ATTENTION

L'écriture sur un secteur absolu est EXTREMEMENT dangereuse car on outrepasse le gestionnaire de fichiers du système d'exploitation.

Exemple :

NEXEMPLE.DAT
W

avec BX=001 CX=03B7, DEBUG affiche la taille du fichier :

Writing 103B7 bytes (Ecriture de 103B7 octets)

puis DEBUG sauvegarde le fichier (EXEMPLE.DAT) sur disque et vous ramène à son signe d'invite lorsqu'il a terminé l'écriture.

Avec :

WCS:100 1 37 2B

DEBUG copie le contenu de la zone mémoire commençant à CS:100 sur la disquette du lecteur B:. L'écriture commence à l'enregistrement logique 37h et touche 2Bh secteurs. DEBUG affiche ensuite son signe d'invite.

12.5 LES MESSAGES D'ERREUR

Durant votre session de travail sous DEBUG, vous pouvez voir apparaître l'un des messages suivants. Chaque erreur interrompt la fonction DEBUG en cours mais ne met pas fin à DEBUG lui-même.

CODE D'ERREUR	DEFINITION
BF	Bad Flag (Mauvais indicateur)

Vous demandiez à modifier l'état d'un flag mais le code tapé ne correspond pas à une valeur correcte. Consultez le tableau de la section consacrée à (R)egister.

BP	Too many BreakPoints (Trop de points d'arrêt)

Dans une commande G, vous avez demandé plus de 10 points d'arrêt. Retapez votre commande G avec 10 points d'arrêt au plus.

BR	Bad Register (Registre incorrect)

Vous avez tapé une commande R en donnant un nom de registre incorrect. Reportez-vous aux explications concernant la commande R.

DF	Double Flag

Vous avez entré deux valeurs pour le même indicateur. Il est bien évident que vous devez n'en donner qu'une.

ANNEXE 13

L'UTILITAIRE EXE2BIN

Cette commande sert à convertir des fichiers exécutables .EXE en fichiers binaires .COM.

SYNTAXE :

EXE2BIN {<lecteur>}<chemin_nom1>{<lecteur>}{<chemin_nom2>}

Cet utilitaire peut donc permettre de convertir des fichiers .EXE en fichiers binaires. Le fichier désigné par <chemin_nom1> est le fichier d'entrée. Si vous ne précisez pas l'extension, .EXE sera choisie par défaut. Le fichier d'entrée est converti au format d'un fichier binaire .BIN qui est une image de la mémoire sur disque. Le fichier résultat prend éventuellement le nom défini par <chemin_nom2>. Si vous ne précisez pas de chemin, le fichier viendra se placer dans le répertoire par défaut, et si vous n'indiquez pas de nom, il aura le nom du fichier d'entrée avec l'extension .BIN.

Le fichier d'entrée doit être un fichier .EXE valide produit par le linker. La partie résidente, c'est-à-dire l'ensemble code et données (data) ne doit pas occuper plus d'un segment, c'est-à-dire 64 K. Il ne doit pas y avoir de segment STACK.

Selon que CS:IP (Code Segment:Instruction Pointer) est ou non spécifié dans le fichier .EXE, EXE2BIN effectue deux sortes de conversions :

1. Si CS:IP n'est pas spécifié par le fichier exécutable, l'utilitaire effectue une conversion binaire pure et simple. S'il faut procéder à des aménagements en ce qui concerne les segments (c'est-à-dire si le programme comprend des instructions impliquant que les segments soient relogés), EXE2BIN vous demandera la valeur à adopter pour ce relogement.

Cette valeur est donc le segment absolu dans lequel le programme devra se charger. Le programme résultant ne sera donc utilisable que s'il est implanté à l'adresse absolue déterminée par une application. L'interpréteur de commandes (COMMAND.COM) ne pourra charger ce programme.

2. Si CS:IP est spécifié comme étant 00:100h, EXE2BIN tient pour acquis que ce programme sera exécutable comme fichier .COM avec une implantation à 100h définie par la pseudo instruction assembleur ORG.

   Les premiers 100h octets du fichier sont supprimés. Aucune manipulation de segment n'est permise étant donné qu'un fichier .COM doit être relogeable par segment. En d'autres termes, il faut se conformer aux spécifications des programmes .COM telles que les explique le manuel du macro assembleur Microsoft. La conversion terminée, vous pouvez renommer le fichier .BIN en fichier .COM (ou affecter directement cette extension dès la ligne de commande). L'interpréteur de commandes (COMMAND.COM) pourra charger ce programme comme il le fait pour tous les programmes .COM figurant sur vos disquettes système.

MESSAGES :

File cannot be converted (Ce fichier ne peut être converti)

CS:IP n'est pas conforme aux spécifications décrites ci-dessus ou le fichier pourrait répondre aux critères définissant un fichier .COM mais tous les segments ne sont pas identiques. Ce message apparaît aussi dans le cas où le fichier n'est pas un fichier exécutable valide.

File not found (Fichier non trouvé)

Le fichier n'est pas dans le répertoire spécifié.

Insufficient memory (Mémoire insuffisante)

File creation error (Erreur durant la création du fichier)

EXE2BIN ne peut créer le fichier résultant. Vérifiez (DIR ou CHKDSK) qu'il y a assez de place sur le disque ou qu'une autre condition particulière n'empêche pas d'écrire le fichier.

Insufficient disk space (Espace disque insuffisant)

Fixup needed - base segment hex : (aménagement nécessaire-segment de base (hex))

Le fichier d'entrée (.EXE) indique qu'il faut spécifier un segment de chargement.

Donnez l'adresse de segment absolue à laquelle le module final doit être relogé.

WARNING - Read error in .EXE file (Attention, erreur de lecture du fichier EXE)

EXE2BIN n'a pu lire le nombre d'octets indiqués par l'en-tête du fichier. Il ne s'agit que d'un message d'avertissement signifiant que l'en-tête ne correspond pas à la taille du fichier.

ANNEXE 14

LA COMMANDE EXIT

La commande EXIT fait sortir du programme COMMAND.COM (l'interpréteur de commandes) et renvoie à un niveau supérieur s'il en existe un.

SYNTAXE :

EXIT

Cette commande sert essentiellement quand, depuis l'intérieur d'une application, vous avez appelé l'interpréteur de commandes MS-DOS (avec lequel vous avez effectué un travail quelconque) et que vous voulez de nouveau retrouver votre application. Par exemple, pour vous permettre de visualiser le catalogue du lecteur B: depuis une application, celle-ci doit appeler la fonction EXEC vers l'interpréteur de commandes (appel système 4Bh). Le signe d'invite MS-DOS apparaît. Vous pouvez utiliser la fonction DIR et COMMAND.COM affiche le catalogue. Vous revenez à votre programme d'application en tapant EXIT en réponse au signe d'invite de COMMAND.COM. Vous repassez ainsi à ce qu'il est convenu d'appeler le niveau supérieur.

ANNEXE 15

L'UTILITAIRE RECOVER

Cet utilitaire permet de récupérer un fichier ou un disque entier présentant des secteurs défectueux.

SYNTAXE :

RECOVER {<lecteur>}

ou

RECOVER <lecteur>{<chemin_nom>}

Si un secteur d'un disque est défectueux, vous pouvez rétablir soit le fichier contenant ce secteur (qui est perdu), soit le disque entier (si le secteur défectueux se trouve dans la zone du catalogue).

Pour réparer un fichier, entrez :

recover <spécificateur_de_fichier>

MS-DOS lit alors le fichier secteur par secteur et saute le ou les secteurs défectueux. Les secteurs présentant un défaut sont marqués par le système d'exploitation qui ne les utilisera plus ensuite pour y écrire des données.

Pour récupérer un disque :

recover <unité>:

où unité correspond à la lette du lecteur.

MESSAGES :

File not found (Fichier non trouvé)

MS-DOS n'est pas parvenu à trouver le fichier désigné. Vérifiez que le chemin donné soit le bon et que le fichier existe bien dans le répertoire spécifié.

(xxxx) of (xxxx) bytes recovered (xxxx octets récupérés sur xxxx)

Ce message indique le nombre d'octets que MS-DOS a pu récupérer sur le disque.

Warning-directory full (Attention : catalogue plein)

Le répertoire racine est trop rempli pour le processus de restauration. Supprimez certains fichiers du répertoire racine pour libérer de la place.

ANNEXE 16

L'UTILITAIRE SHARE

L'utilitaire SHARE (partager) met en place le partage et le verrouillage des fichiers.

SYNTAXE :

share {/f:<espace>}{/L<verrous>}

La commande SHARE n'a de sens que dans un environnement de réseau local actif. Elle est destinée à figurer dans AUTOEXEC.BAT pour installer le partage des fichiers. Reportez-vous au Microsoft Networks Manager's Guide si vous désirez approfondir votre connaissance des réseaux (LANs).

Le commutateur /f:<espace> permet de préciser l'espace mémoire que pourra prendre MS-DOS (versions 3.1 et au dessus) pour l'enregistrement des informations concernant le partage. Chaque fichier partagé y figure et occupe un nombre d'octets égal à la longueur du nom de fichier plus 11 (ce qui donne une moyenne de 20 octets). La valeur par défaut pour /f est 2048 octets.

Le commutateur /l permet de préciser le nombre de blocages de régions autorisé. La valeur par défaut est 20.

Par exemple, la commande suivante appelle le partage des fichiers (en fait la gestion des conflits d'accès) et utilise les valeurs par défaut des options /f et /l :

share

MESSAGES :

Incorrect parameter (Paramètre incorrect)

Une des options spécifiées est fausse.

Not enough memory (Pas assez de mémoire)

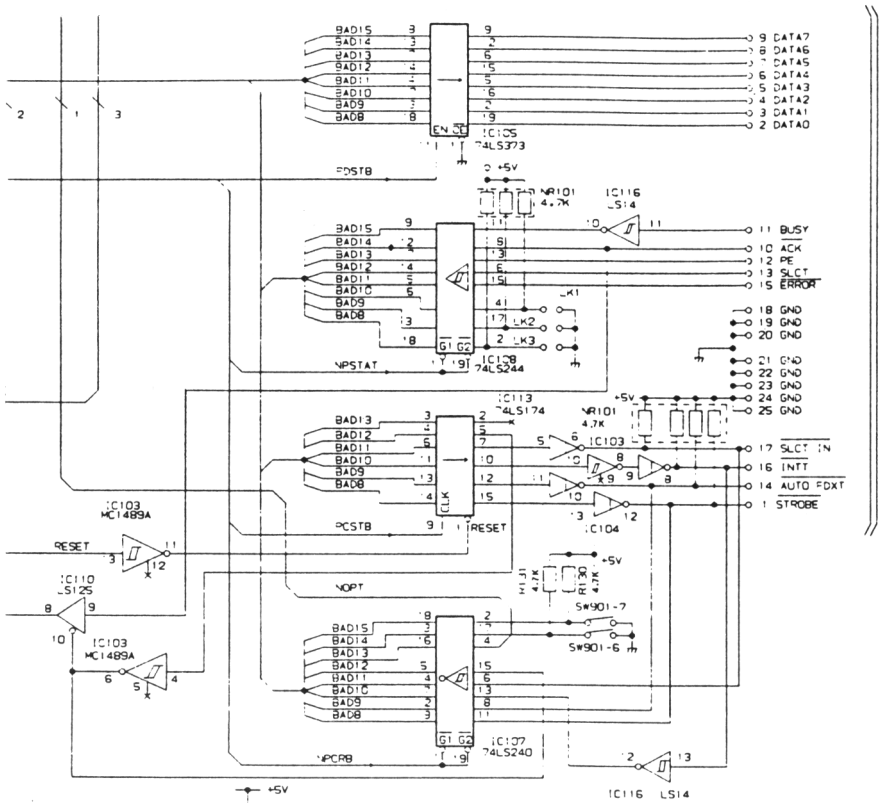
MS-DOS ne dispose pas d'assez de mémoire pour lancer la commande.

Share already installed (Partage déjà installé)

On ne peut installer SHARE qu'une fois.

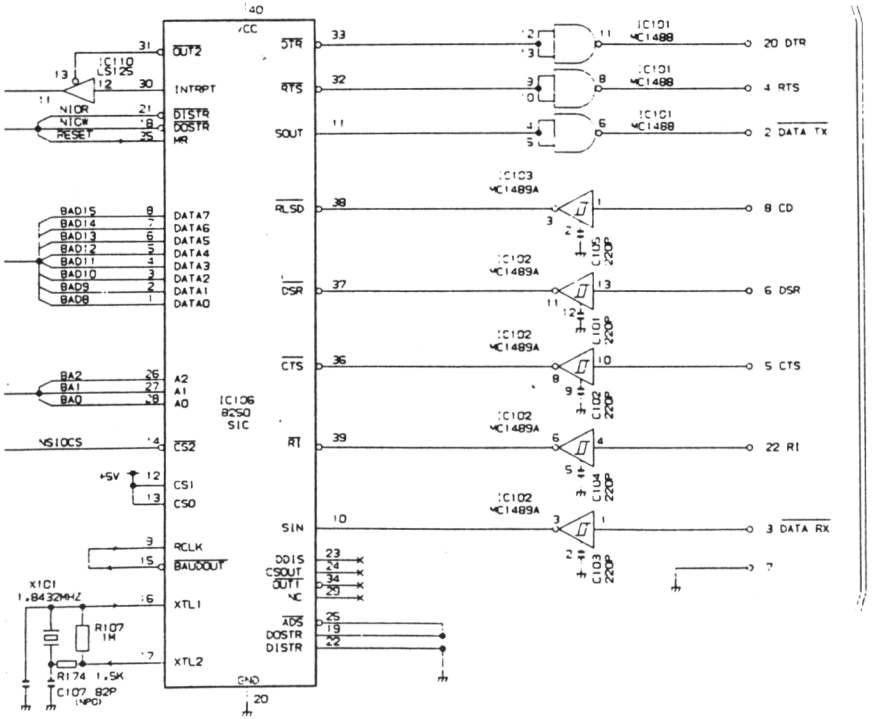
ANNEXE 17

SCHEMAS DES PORTS IMPRIMANTE ET SERIE



JD 102 PRINTER 25 WAY D-SUB CONN.

Schéma du port imprimante



JP 101 SERIAL PORT 25 WAY D-SUB CONN.

Schéma du port série

INDEX

1234h 133-219
24 heures (mode) 203-218
32,768 Khz (oscillateur) 22
320x200 (graphisme) 34
640x200 (mode graphique) 34
640x350 (mode graphique) 34
8087-2 (co-processeur numérique) 3-180

A

ACE (registres 8250) 190
Affichage couleur en mode texte 42-49
Affichage graphique couleur 42
Alarme RTC (inhibition) 208
Alimentation (connecteur) 119
Allemand (clavier) 317
Allemands (messages du ROS) 227
Amstrad PC 1640 (message de version) 224
Attribute Controller Address (registre) 42-47
Autotest clavier et souris 224

B

Basse résolution (graphiques 320x200) 34
Baud 190
BIOS IGA (Initialisation mode EGA) 84
Bootstrap (interruption) 202
BREAK (commande) 287
Break (octet) 129-192
BUFFERS (commande) 288

C

Câble Null modem 306
Canaux d'E/S de la carte mère 7
Caractère (Plan Mapping) 64
Carte mémoire 4

Centronics (port compatible) 23
CGA (adressage du contrôleur vidéo) 89
CGA (Color Select Register) 88
CGA (mode compatible, registres) 89
CGA (mode émulation 6845) 90
CGA (status register) 89
Character Set Select (Registre) 54
Chargement des têtes (délai) 210
Checksum (erreur sur ROM externe) 133-225
Clavier (autotest) 137
Clavier (codes clavier) 315
Clavier (connecteur) 109
Clavier (disposition des touches) 317
Clavier (état du tampon) 197
Clavier (interface) 108
Clavier (interruption break) 207
Clavier (interruption d'E/S) 137
Clavier et codes de touches 227-315
Clock Mode (Registre) 202
Cluster (contrôleur externe) 9
Code touche absolu 141-214
Color No Care (Registre) 65
Color Plane Compare (Registre) 59
Color Plane Enable (Registre) 50
Color Plane Read (Registre) 62
Color Plane Select (Registre) 63
COMMAND.COM 294
Compteur 1 (initialisation) 15
Contrôleur d'interruptions (initialisation) 13
Contrôleur d'interruptions programmable (test) 137
Contrôleur de disquettes 102
Contrôleur disque (erreur) 181-184
Contrôleur graphique (registres) 57
CPU 3
Crayon optique (connecteur) 114
Crayon optique (obtenir position) 152-164
Crayon optique (regitres) 77-78
CRC (erreur) 182
CRTC (données d'initialisation) 163
CRTC Mode (registre) 81

CRTC Offset (registre) 79
CRTC Overflow (registre) 71
CTRL [touches] (action des touches) 144
Curseur (départ) 217
Curseur (obtenir position) 152-164
Curseur (positionnement) 151
Curseur (taille) 151
Cursor End (octet) 217
Cursor End (registre) 74
Cursor Location High (registre) 75
Cursor Location Low (registre) 76
Cursor Start (registre) 73

D

Danois (clavier) 320
Danois (messages du ROS) 227
Data Rotate (Registre) 60
DDM 18-180
DEBUG 351
Détermination du type PC 1640/PC 1512 24
DEVICE (commande) 289
Disk I/O (interruption) 181-183
Disque dur (appels de paramètres et registres) 186
Disque dur (interruption d'E/S) 181-186
Disque dur (nombre) 218
Disque dur (ROM) 3-4-186-229
DMA (Autotest du contrôleur) 135
DMA (contrôleur 8237-A Registre) 235
DMA (erreur d'overrun) 181
DMA (erreur pour dépassement des 64 K) 181-184
DMA (initialisation) 11
DMA 9
DMA Page (registres) 10
Drive Motor (compteur) 216
Drive Motor (flag) 216
Drive Restore (flag) 215
Drive Status (octet) 216
DRIVER.SYS 290
DRIVPARM (commande) 292

E

E/S disques (interruption) 181
ECC (erreur) 186
Ecriture (émulation TTY) 159
Ecriture (palette vidéo) 169
Ecriture (pixel Int vidéo 16) 158-169
Ecriture (secteur) 183
Ecriture de caractère et d'attribut 155-168
Ecriture de caractère seul 156-169
Editeur de liens 323
EGA (Horizontal Display End, registre) 67
EGA (Horizontal Total, registre) 67
EGA (registre Address) 67
EGA (registres de contrôle) 66
EGA CRTC (mode vidéo compatible) 161
EGA CRTC (registre Mode) 81
EGC (Registre control) 44
EGC (Registre External Control) 44
End Horizontal Blanking (registre) 68
End of Track 210
End Vertical Blanking (registre) 81
Envoi de caractère vers le port imprimante 199
Erreur disque en écriture 181
Espagnol (clavier) 319
Espagnol (messages ROS) 227
EXE2BIN (utilitaire) 383
EXIT (commande) 387
Extended Graphics Border (registre) 49
Extended Mode Control (registre) 39
Extension (bus) 7
Extension (cartes) 116
Extension (connecteurs de cartes) 116
External Ticker (interruption) 208

F

FCBS (commande) 293
FDC (Mots de commande) 255
FDC (mots de commandes invalides) 269
FDC (uPD765A, registres) 250

FILES (commande) 294
Fonctions avancées (interruption) 194
Fontes (de la ROM IGA) 4-229
Formatage (disque) 186
Formatage (piste défectueuse) 186
Formatage (piste) 184-186
Français (clavier) 317
Français (Messages du ROS) 227

G

Gap (longueur) 210
Graphics Mode (Registres 1 et 2) 62-63
Grec (clavier) 321

H

Haut parleur 16
Haute résolution (mode 640x350) 37
Hercules (registres) 41
Hercules 6845 CRTC (émulation) 100
Horizontal Total 67-92-100-208

I

IGA (BIOS) 37
IGA (registres de contrôle) 38
Imprimante (câblage) 309
Imprimante (état) 26
Imprimante (interruption d'E/S) 199
Imprimante 23-26
Imprimantes (nombres) 180
Initialisation (autotests) 126
Initialisation (pile) 211
Initialisation (sous-système disque) 182-186
Initialisation des vecteurs d'interruptions 126
Initialisation port imprimante 199
Initialisation port série 189
Interface de périphériques programmable (test) 135
Interface parallèle 106

Interfaces série (nombres) 180
Internal Graphics Adapter 27
Interpréteur de commandes 349
Interruption 14 : Contrôleur de disquettes 147
Interruption 16 : E/S vidéo 147
Interruption 16 en mode compatible EGA 161
Interruption 17 : Configuration système 179
Interruption 18 : Taille mémoire 180
Interruption 19 : E/S disques 181
Interruption 2 : erreur de parité 138
Interruption 20 : E/S série 189
Interruption 21 : Fonctions avancées 194
Interruption 22 : E/S clavier 197
Interruption 23 : E/S imprimante 199
Interruption 24 : Redémarrage système 201
Interruption 25 : Bootstrap 202
Interruption 26 : Horloges 202
Interruption 27 : Break clavier 207
Interruption 28 : Ticker 208
Interruption 29 : Table des paramètres vidéo 208
Interruption 30 : Table des paramètres disques 210
Interruption 31 : Matrice vidéo 210
Interruption 5 : Impression d'écran 138
Interruption 51 : E/S souris 271
Interruption 6 : Contrôle des boutons souris 139
Interruption 8 : horloge système 140
Interruption 9 : Clavier 141
Italien (clavier) 319
Italiens (Messages du ROS) 227

J

Jeu de caractères (Sélection A/B) 30-54-55
Joystick (code clavier des boutons) 221
Joystick (connecteur) 114
Joystick (interface) 113

L

LASTDRIVE (commande) 294
Lecture caractères et attributs 154
Lecture de la NVR 195-196
Lecture pixel (Int vidéo 16) 159
Lecture port série 189-191
Lecture secteur 182
Liaisons du langage 227
LIB (Bibliothèques) 328
Line compare (registre) 82
Linker 323

M

MAP 329
Marqueur de secteurs défectueux 188
Matrice de caractères (chargement) 174
Maximum Scan Line (registre) 72
MC6845 (modes compatibles) 147
MDA (registres) 93
Memory mode Select (registre) 56
Messages ROS 27-223-227
Mise en route (processus) 135
Mode autotest 27-227
Mode Control (registre) 48
Mode vidéo courant (octet) 216
Mode vidéo courant 191
Mode vidéo par défaut 18-121-180
Moyenne résolution (640x200) 36
MS-DOS configùration 287
MS-LINK 323

N

NDP 3-180
Niveaux d'interruptions 13-14-126
NMI (masque de contrôle) 13
NMI Inhibée 16
Norvégien (clavier) 322
NVR (Mode série initial) 221

O

OBJ 328
Octet d'attribut (couleur) 31
Octet d'attribut (monochrome) 33
Octet de remplissage 210
Octet de sélection de couleur 191
Overscan (registre) 49

P

Page d'écran (active) 154
Page vidéo active 217
Palette (registres) 28-48
Paper Out 199
Paramètres d'heure et date 221
Paramètres Vidéo (obtenir) 161-173
Parity Error (NMI) 138
Pays (sélections relatives au) 297
PIC (mots de commande) 238
PIT (registres) 242
Plantronics 41
Port A 17
Port B 16
Port C 19
Portugais (clavier) 321
Positionnement des têtes (délai) 210-211
Preset Row Scan (registre) 72
Print Screen (état) 220
Print Screen 138
Printer Ackowledge 199
Printer control (latch) 24
Printer data (latch) 23
Printer Idle 199

R

RAM (taille installée) 19
RAM Dynamique (rafraichissement) 15
RAM Non Volatile 221
RAM système (test) 136

RAM0-RAM4 19
RAMDRIVE.SYS Compteurs X et Y (lecture) 291
RECOVER (utilitaire) 389
Répartition électrique 311
Reset (soft) 144
Reset système 21
Resident Operating System 3
ROM (jeux de caractères) 313
ROS (erreurs fatales) 225
ROS 125
RS232 299
RTC (obtenir date) 205
RTC (obtenir heure) 204
RTC (test) 135
RTC 22

S

Scrolling 154-165
Seek error 181
Sélection AND, OR, XOR vidéo 60
Sélection de couleur IRGB 27-29
Sequencer (registres) 52
Sequencer Address (registre) 52
Série (brochage) 105
Série (connecteur) 104
Série (interruption d'E/S) 189
SHARE 391
SHELL 294
Souris (codes clavier) 221
Souris (connecteur) 111
Souris (curseur) 271
Souris (interface logicielle) 110
Souris (interruption) 139
Souris (test des registres) 136-221
STACKS (commande) 295
Start Adress High (registre) 74
Start Adress Low (registre) 75
Start Horizontal Blanking (registre) 68
Start Horizontal Retrace (registre) 70

Start Vertical Blanking (registre) 80
Start Vertical Retrace (registre) 76
Status port (registre) 83
Suédois (clavier) 320
Suédois (messages ROS) 227
SW1-SW4 (lecture d'état) 46
SW6, SW7, SW9 et SW10 (lecture d'état) 24
Switches (Sélection de l'affichage) 121
Système (horloge) 202
Système (interruption de configuration) 180
Système (mémoire) 3
Système d'exploitation 125

T

Table des paramètres disque 210
Taille secteur 210
Tampon (clavier) 215
Tampon emplacement curseur 187
Test du processeur de communication asynchrone 133
Timeout moteur off 210
Timer (configuration) 15
Touches (adresses) 220
Touches (bascules) 197-211
Touches (état) 197-211
Touches (Key codes) 109-141-229-315
Touches (tokens) 141
Touches spéciales 144

U

Underline Location (registre) 79
USA (clavier) 318

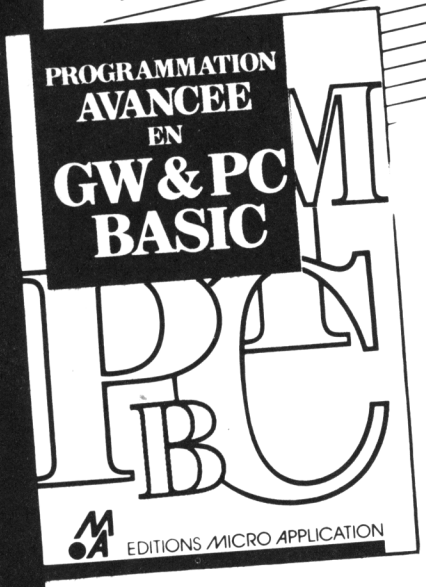
V

Variables système 211
Vérification de secteurs 183
Vertical Display End (registre) 78
Vertical Interrupt (IRQ2) 78

Vertical Total (registre) 71
Vidéo (connecteur) 118
Vidéo (E/S) 211
Vidéo (matrice) 210
Vidéo (table des paramètres) 208
Video Status MUX 50
Vidéo Tampon 33-34
Vitesse de transmission 187
VM.TMP 329

W

Wait States 3



LE LIVRE DU GW BASIC & PC BASIC

Une approche complète et didactique de la programmation en GW/PC Basic. Découvrez toutes les possibilités de ce langage. La syntaxe et les différentes possibilités de paramétrage sont expliquées en détail : instructions et fonctions, gestion de fichiers, programmation sonore, graphismes, technique des fenêtres, interface, programmation par interruptions, compilation et programmes utilitaires à saisir. Réf : ML 170. 149 F. 327 p.

BEST

BIEN DEBUTER SUR PC

Réussissez à coup sûr vos débuts et évitez les erreurs. Apprenez à connaître tout d'abord votre nouveau matériel (écran, clavier, unité centrale ...), puis à bien utiliser le DOS et toutes ses commandes. Une initiation complète au Basic est également fournie vous permettant de commencer à programmer votre tout nouveau PC. Réf : ML 183. 149 F. 327 p.

ECRANS ET FICHIERS EN LANGAGE C

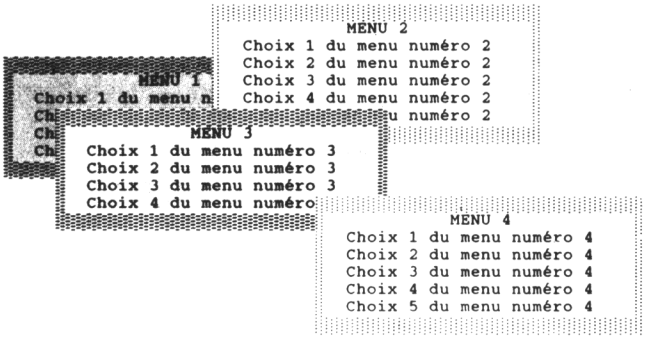
Comment utiliser le langage C et réaliser une gestion d'écran conviviale (fenêtres, menus déroulants...) et un système de gestion de fichiers (accès direct, séquentiel...). Acquisition des données : menus et écrans, structure de l'écran IBM PC et compatibles, utilisation des interruptions du BIOS, réalisation d'une fenêtre (ouvrir, déplacer, agrandir, zoom), gestion des menus déroulants, technique de multi-fenêtrage. Stockage des données : différentes fonctions pour gestion de fichiers en C, fichiers accès séquentiel, et fichiers accès direct. Bien sûr, tous les utilitaires fournis peuvent être intégrés dans vos programmes. Réf : ML 182. 199 F. 300 p. Réf : ML 282. 299 F le livre et la disquette.

PROGRAMMATION AVANCEE EN GW & PC BASIC :

Ce livre est la suite logique du LIVRE DU GW BASIC & PC BASIC. Il s'adresse à tous ceux qui maitrisent le GW Basic et qui veulent réaliser des programmes de qualité professionnelle. Les domaines traités sont les tris, la gestion d'écran, les fichiers et les techniques d'accès aux données, la gestion des imprimantes etc. Le livre comprend également de très nombreux exemples, trucs et programmes utilitaires qui se révèleront très utiles une fois intégrés dans vos propres programmes. Réf : ML 190. 199 F. 446 p. Réf : ML 290. 319 F le livre et la disquette.

DU BASIC AU TURBO PASCAL

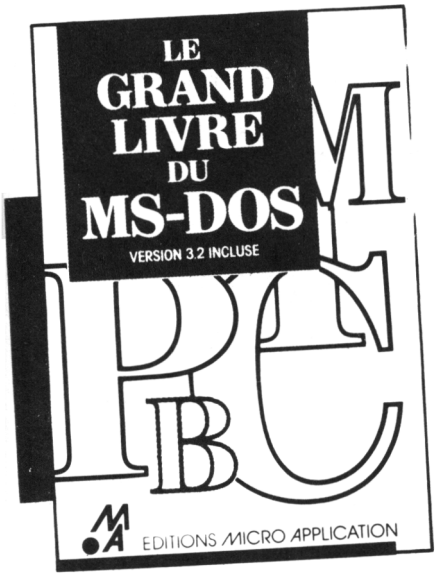
Le Turbo Pascal est le langage standard de programmation sur PC. Vous qui maîtrisez déjà le BASIC et qui voulez programmer en Turbo Pascal, prenez rapidement connaissance des commandes de l'éditeur et des options de compilation, apprenez les bases de la programmation en Pascal et comment développer sous Turbo des routines correspondant à des programmes Basic. Ainsi votre investissement Basic n'est pas perdu et vos futurs développements profiteront des qualités du Turbo Pascal comme la programmation structurée et la rapidité d'exécution. Réf : ML 186. 199 F. 305 p.



```
else
 begin
  if x = b^.ident then ok := true
   else
    begin
     c := b;
     if x < b^.ident  b := b^.gauche
      else
       b := b^.droite
    end
```

TRUCS ET ASTUCES POUR TURBO PASCAL

Grâce aux nombreux conseils, méthodes, programmes et utilitaires présentés dans ce livre, utilisez de façon optimale le langage Turbo Pascal sur votre PC. Disposez de nombreux utilitaires pour faciliter vos développements sur Turbo Pascal, et améliorez la productivité de vos applications avec les procédures spécialement étudiées pour s'intégrer à tous vos programmes. Réf : ML 133. 149 F. 256 p. Réf : ML 233. 269 F le livre avec la disquette.



LE GRAND LIVRE DU MS/DOS

Pour bien utiliser son PC, il est essentiel de bien connaître son système d'exploitation le MS/DOS et ses nombreuses commandes. Cet ouvrage didactique et pratique comporte une introduction pour les débutants et couvre toutes les commandes du DOS jusqu'à la version 3.2. Réf : ML 192. 149 F. 358 p.

LE C FACILE

Votre premier programme en C, les types de données, les attributs, adresses et pointeurs, la structure générale des programmes et l'organisation interne de C... Evitez avec ce livre les fautes typiques. Très utile : du Basic et du Pascal au C, comment convertir facilement vos programmes et une description des compilateurs. Réf : ML 191. 149 F. 270 p.

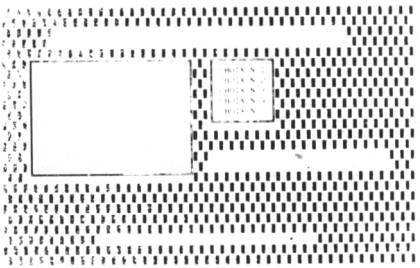
LE LANGAGE MACHINE SUR PC

Réussissez vos débuts en assembleur. Plus qu'un simple manuel, ce livre contient une partie référence : commandes assembleur, instructions et fonctions du macro assembleur, routines BIOS et DOS... Et très important : l'interfaçage de routines assembleur avec d'autres langages (C, Turbo Pascal, Basic...). Réf : ML 189. 199 F. 580 p

A PARAITRE

LA BIBLE DU TURBO PASCAL

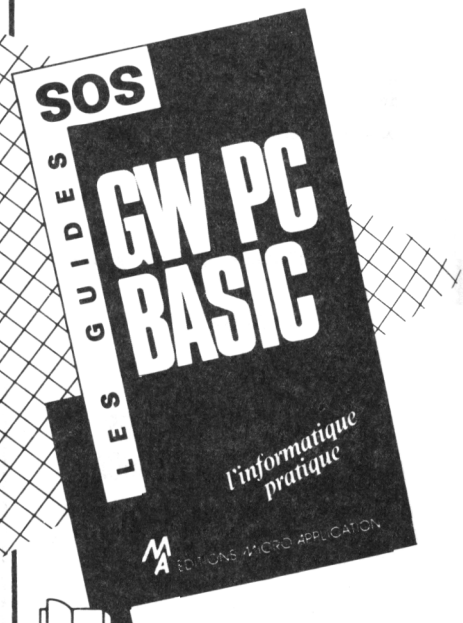
Tout pour la programmation professionnelle en TURBO PASCAL. Cet ouvrage comprend une introduction pratique au Turbo Pascal et aux bases de MS-DOS/PC-DOS à la documentation des programmes, en passant par des conseils et trucs pour une programmation efficace et rationnelle. Plusieurs projets sont proposés ainsi que de nombreux programmes qui constituent une bibliothèque très utile. Réf : ML 187. 249 F. 700 p. Réf : ML 287. 399 F le livre avec 2 disquettes.



L'ÉNERGIE MICRO

LIVRES PC

EN PLEINE PROGRAMMATION, OU BIEN PENDANT L'UTILISATION D'UNE APPLICATION SUR VOTRE ORDINATEUR, LE TROU, LE VIDE TOTAL. COMMENT S'APPELAIT DONC CETTE INSTRUCTION POUR CREER UNE PARTITION SUR DISQUE DUR ? IMPOSSIBLE DE S'EN SOUVENIR ! UNE RECHERCHE DANS LE MANUEL S'AVERE LONGUE ET FASTIDIEUSE. C'EST A CET INSTANT QU'INTERVIENT LE GUIDE SOS MICRO APPLICATION : TOUTES LES INSTRUCTIONS SONT CLASSEES PAR RUBRIQUE ET ORDRE ALPHABETIQUE, AVEC LES PARAMETRES NECESSAIRES ET UNE BREVE EXPLICATION. AINSI, QUELLE QUE SOIT LA FORME SOUS LAQUELLE LE PROBLEME SE PRESENTE, UN BREF COUP D'ŒIL SUR LE GUIDE ET IL SERA RESOLU.



GUIDE SOS GW/PC BASIC

Ce livre expose sous une forme claire et pratique toutes les spécificités du langage GW/PC BASIC. Vous y trouverez, organisés selon un ordre logique, la syntaxe de toutes les commandes GW/PC Basic, la liste des fonctions de l'éditeur, les messages d'erreur, gestion des interruptions, etc... Les multiples index et répertoires du guide SOS GW/PC Basic vous aident à trouver immédiatement l'information que vous désirez.
Réf. GL 104. 129 F. 200 p.

GUIDE SOS TURBO PASCAL

C'est le guide de référence pour tous les programmeurs en Turbo Pascal. Il est organisé en plusieurs sections qui traitent chacune d'un thème spécifique : l'éditeur (menus, codes de contrôle...), les commandes (syntaxe et description, graphiques, entrées/sorties...), les structures de contrôle, les appels MS-DOS ou CP/M, l'édition de liens avec asssembleur... Grâce aux index et répertoires faisant référence aux mots clefs de Turbo Pascal, vous retrouverez immédiatement l'information dont vous avez besoin.
Réf. GL 106. 99F. 150 p.

GUIDE SOS TURBO C

Destiné au programmeur en langage C, ce guide décrit en détail toutes les commandes et instructions du langage Turbo C. Pour chaque instruction la syntaxe et une description précise des fonctions est présentée et une petite routine explique la programmation. Les nombreux répertoires et index du guide SOS Turbo C vous permettent de retrouver très rapidement l'information que vous désirez. (Liste détaillée des commandes, fonctions et instructions par ordre alphabétique). Réf. GL 107. 149 F. 350 p.

A PARAITRE

GUIDE SOS dBASE III/dBASE III+

Conçu comme étant un outil de travail pour l'utilisateur, le GUIDE SOS dBASE III/dBASE III+ présente de façon pratique et explicite l'ensemble des commandes et fonctions des logiciels dBASE III et dBASE III+ (exploitation de dBASE III+ en réseau...). Il offre plusieurs index et répertoires pour retrouver très rapidement l'information désirée. Réf. GL 108. 99 F. 170 p.

GUIDE SOS GEM APPLICATIONS

Il s'adresse aux utilisateurs des logiciels conçus pour l'intégrateur GEM de DIGITAL RESEARCH. Il montre ainsi comment tirer parti de GEM DRAW, GEM WRITE, GEM PAINT, GEM WORD CHART... De plus, il explique comment transférer des données entre les différents programmes sous GEM, et donne quelques trucs et astuces pour l'exploitation de ces logiciels.
Réf. GL 109. 149 F. 300 p.

GUIDE SOS MULTIPLAN

Il constitue une source d'information très pratique et bien documentée pour l'utilisateur du logiciel MULTIPLAN (version 3.0, Junior ou CP/M). Organisé autour de plusieurs index de références et sommaires, le GUIDE SOS MULTIPLAN décrit toutes les commandes de MULTIPLAN 3.0, l'exploitation de MULTIPLAN en réseau, le transfert de données entre MULTIPLAN et les autres tableurs (VISICAL, LOTUS 1- 2-3, SYMPHONY) et l'installation d'imprimantes. Réf. GL 110. 99 F. 150 p.

GUIDE SOS MS-DOS

Outil de référence pour tout utilisateur du système d'exploitation MS-DOS 3.3 (et versions antérieures) cet ouvrage est présenté sous un format facilitant l'accès à toutes les informations utiles. Ce guide décrit d'une part la syntaxe des commandes MS-DOS 3.3 et, d'autre part l'exploitation des fonctions (fichiers Batch, fichier Config. SYS,...). Réf. GL 105. 129 F. 210 p.

GUIDE SOS WORD

C'est un outil de travail pour les logiciels de traitement de texte WORD 3.0, WORD 2.0 et WORD JUNIOR. Structuré de manière à faciliter la recherche d'informations, le GUIDE SOS WORD analyse en détail chaque aspect de ces logiciels (commandes, menus, fonctions, correction orthographique et calcul sur WORD 3.0...) et les expose de façon claire et pratique. Réf. GL 111. 129 F. 200 p.

GUIDE SOS NORTON UTILITIES

Les NORTON UTILITIES sont un ensemble de programmes utilitaires très pratiques pour connaître le fonctionnement interne de votre PC, XT ou AT pour rattraper les erreurs malencontreuses (suppression de fichier par exemple). Il vous explique comment bien exploiter les NORTON UTILITIES versions 3.0 et 4.0. Chacun des 14 programmes est présenté en détail à l'aide d'exemples d'applications. De plus le GUIDE NORTON UTILITIES consacre un chapitre pour chaque programme « NORTON UTILITY » et « NORTON INTEGRATOR ».
Réf. GL 112. 129 F. 200 p.

LE GUIDE SOS PROGRAMMATION DE L'AMSTRAD PC

Réf. GL 101. 149 F. 320 p. Voir informations détaillées p. 7.

GUIDE SOS FIRST WORD

Réf. GL 103. 129 F. 250 p.
Voir informations détaillées p. 15.

GUIDE SOS GFA BASIC

Réf. GL 102. 149 F. 300 p.
Voir informations détaillées p. 15.

Achevé d'imprimer
sur les presses de l'imprimerie IBP
à Rungis (Val-de-Marne 94) (1) 46.86.73.54
Dépôt légal - Février 1988
N° d'impression: 4979

Le guide de référence technique du PC 1640 (PC 1640 Firmware Book).

Attention, vous qui désirez tout savoir sur le PC 1640 ce guide a été spécialement écrit pour vous par l'équipe développement de chez AMSTRAD INTERNATIONAL. Il vous dévoilera toutes les caractéristiques de l'AMSTRAD PC et vous donnera les informations nécessaires à une programmation poussée de cet ordinateur. Parmi les différents sujets abordés, vous trouverez :

- L'unité centrale
- Organisation de la mémoire
- Le DMA
- Le BIOS IGA
- Les interruptions systèmes
- Etat et Contrôle du système
- Horloge temps réel
- Contrôleur VDU et IGA couleur alpha / graphique
- Contrôleur de disquette (FDC)
- Port sériel RS232C
- Port parallèle d'imprimante
- Interfaces et connexions (clavier, souris, joystick...)
- Initialisation et Tests à la mise sous tension
- Interruptions ROS
- RAM non volatile
- Message ROS
- Les utilitaires MSLINK et DEBUG
- Nombreuses informations de références
- Annexes détaillées

Réf. : ML 511
Prix : 249 Francs TTC

AMSTRAD

PC 1640 GUIDE DE REFERENCE TECHNIQUE